



NOVA

IMS

Information
Management
School

MGI

Mestrado em Gestão de Informação

Master Program in Information Management

From User Browsing Behaviour to User Demographics

Filipe Manuel Teixeira Lopes Pereira

Thesis presented as partial requirement for the degree of
Master of Information Management

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

NOVA Information Management School

Universidade Nova de Lisboa

**From User Browsing Behaviour
to User Demographics**

Filipe Manuel Teixeira Lopes Pereira

Thesis presented as partial requirement for the degree of Master of Information Management, specialisation in Information Systems and Technologies Management

Supervisor: Mauro Castelli

Co-supervisor: Leonardo Vanneschi

October 2015

Acknowledgements

I would like to express my sincere gratitude to my supervisor Prof. Mauro Castelli for the continuous support of my Master study and for his patience, motivation and knowledge. I could not have imagined a better advisor and mentor for my Master thesis. The same goes for my co-supervisor Prof. Leonardo Vanneschi for his support throughout the writing of the thesis.

I would also like to thank my parents, brothers and girlfriend for supporting me throughout my studies and life in general. A special thanks to my brother João, for his help with ensuring the coherence and consistency of the text.

Resumo

A Internet conta hoje com mais de 3 mil milhões de utilizadores e esse valor não para de aumentar. Desta forma, proporcionar uma experiência online agradável aos seus utilizadores é cada vez mais importante para as empresas. De modo a tirar partido dos benefícios deste crescimento, as empresas devem ser capazes de identificar os seus clientes-alvo dentro do total de utilizadores; e, subseqüentemente, personalizar a sua experiência online. Existem diversas formas de estudar o comportamento online dos utilizadores; no entanto, estas não são ideais e existe uma ampla margem para melhoria.

A inovação nesta área pode comportar um grande potencial comercial e até ser disruptiva. Com isto em mente, proponho-me a estudar a possível criação de um sistema de aprendizagem automática (“machine learning”) que permita prever informações demográficas dos utilizadores estritamente com base no seu comportamento online. Tal sistema poderia constituir uma alternativa às atuais opções, que são mais invasivas; mitigando assim preocupações ao nível da proteção de dados pessoais.

No primeiro capítulo (Introdução) explico a motivação para o estudo do comportamento dos utilizadores online por parte de empresas, e descrevo as opções disponíveis atualmente. Apresento também a minha proposta e o contexto em que assenta. O capítulo termina com a identificação de limitações que possam existir *a priori*.

O segundo capítulo (Machine Learning) fornece uma introdução sobre machine learning, com o estudo dos algoritmos que vão ser utilizados e explicando como analisar os resultados.

O terceiro capítulo (Implementação) explica a implementação do sistema proposto e descreve o sistema que desenvolvi no decorrer deste estudo, e como integrá-lo em sistemas já existentes.

No quarto capítulo (Análise e manipulação dos dados), mostro os dados compilados e explico como os recolhi e manipulei para testar a hipótese.

No quinto capítulo (Análise de dados e discussão) vemos como é que os dados recolhidos foram usados pelos vários algoritmos para descobrir como se correlacionam com dados dos utilizadores e analiso e discuto os resultados observados.

Por fim, o sexto e último capítulo apresenta as conclusões. Dependendo dos resultados, mostro como a hipótese poderia ser melhor testada, ou então discuto os próximos passos para tornar o sistema realidade.

Palavras-chave

Comportamento online dos utilizadores de internet; Classificação de utilizadores; Previsão de informações demográficas; Aprendizagem automática (Machine Learning)

Abstract

With over 3 billion internet users and counting, providing an enjoyable online experience is becoming increasingly important for businesses. In order to reap the benefits of this growth, businesses need to be able to identify their target customers amongst total users; and then adjust their online presence accordingly. There are multiple ways of learning about online users' behaviour; however, they are sub-optimal and there is ample margin for improvement.

Innovation in this area could yield a vast commercial potential and possibly even be disruptive. With this in mind, I propose to study the possible creation of a machine learning system that would allow businesses to predict demographic and related information about users based solely on their browsing behaviour. Such a system could constitute a valid alternative to current standard practice which is more invasive, therefore raising data privacy concerns.

In chapter 1 (Introduction) I explain the motivation for businesses to gather users' browsing behaviour and describe the options currently available to them. I also explain my proposal and present the technical background that underpins it. I end the chapter with remarks on limitations I know to exist *a priori*.

Chapter 2 (Machine learning) provides an introduction to machine learning, with a study of the algorithms that were used and how to analyse their results.

Chapter 3 (Implementation) deals with the implementation of the proposed system. I describe which component of the system I have developed while working on the present thesis and explain how to integrate it within existing businesses.

In chapter 4 (Data Gathering and Manipulation), I display the data that I have gathered from online users to test the hypothesis put forward. I also show which tools I have used to gather the data and how I manipulated it to be tested.

In chapter 5 (Data analysis and Discussion) I explain how the data gathered was used with several machine learning algorithms to find correlations with users' demographics and analyse and discuss the observed results.

Chapter 6 (Conclusion) focuses on drawing conclusions. Depending on the results, I either show how the hypothesis can be better tested, or otherwise talk about the next steps in order to turn the system into a reality.

Keywords

Online User Behavior; User Profiling; Demography Prediction; Machine Learning

Contents

1	Introduction	1
1.1	Current Solutions	1
1.1.1	Online Marketing	2
1.1.2	Customer input	2
1.1.3	Online analytics tools	3
1.2	Background	3
1.3	Proposal	6
1.3.1	Integration	7
1.4	Further possibilities	8
1.5	Limitations	8
1.5.1	Data and system limitations	8
1.5.2	Ethics	9
2	Machine Learning	10
2.1	Designing a learning problem	11
2.2	Supervised and unsupervised learning	12
2.3	Classification	12
2.3.1	Radial basis function network (RBFNetwork)	13
2.3.2	Bayesian network	14
2.3.3	Rotation forest	15
2.3.4	Analysis	17
2.4	Regression	20
2.4.1	Gaussian processes	21
2.4.2	Sequential minimal optimization for regression (SMOreg)	23
2.4.3	Analysis	24
2.5	Cross-validation	25
2.6	Weka	25

3	Implementation	27
3.1	Database models	27
3.1.1	Device Model	28
3.1.2	Session Model	29
3.1.3	Page Model	29
3.1.4	Interval Model	30
3.2	Software	30
3.2.1	Events	30
3.2.2	Information tracked	31
3.2.2.1	Device information	31
3.2.2.2	Page information	32
3.2.2.3	User browsing information	32
3.2.3	Server side app	32
3.2.3.1	Init event handling	33
3.2.3.2	Interval event handling	33
3.2.3.3	Session validity definition	33
3.2.4	Frontend library	34
3.2.4.1	Init event handling	34
3.2.4.2	Init interval handling	34
3.2.5	Machine learning system	34
4	Data gathering and manipulation	35
4.1	How the data was gathered	35
4.2	How the users were obtained	35
4.3	Data gathered	36
4.4	Data manipulation	37
5	Data analysis and Discussion	39
5.1	The presence of noise	41
5.2	Age	42
5.2.1	Classification	43
5.2.1.1	RBFNetwork	43
5.2.1.2	Bayesian network	43
5.2.1.3	Rotation Forest	43
5.2.1.4	Analysis	44
5.2.2	Regression	45
5.3	Gender	45

5.3.1	RBFNetwork	45
5.3.2	Bayesian network	46
5.3.3	Rotation Forest	46
5.3.4	Analysis	46
5.4	Education	47
5.4.1	RBFNetwork	47
5.4.2	Bayesian network	48
5.4.3	Rotation Forest	48
5.4.4	Analysis	48
5.5	Conclusion	49
6	Conclusion	50
6.1	Future steps	50
A	Repository	52
B	Events	53
C	Questionnaire screenshots	56
D	Facebook ads	62
E	Data gathered	63
F	Scripts for data manipulation	64
G	Classification on age	65
G.1	CVParameterSelection for the RBFNetwork model	65
G.2	RBFNetwork model	66
G.3	Bayesian network model	67
G.4	CVParameterSelection for the Rotation Forest model	68
G.5	Rotation Forest model	69
H	Classification on gender	70
H.1	CVParameterSelection for the RBFNetwork model	70
H.2	RBFNetwork model	71
H.3	Bayesian network model	72
H.4	CVParameterSelection for the Rotation Forest model	73
H.5	Rotation Forest model	74

I	Classification on education	75
I.1	CVParameterSelection for the RBFNetwork model	75
I.2	RBFNetwork model	76
I.3	Bayesian network model	77
I.4	CVParameterSelection for the Rotation Forest model	78
I.5	Rotation Forest model	79
J	Regression on age	80
J.1	Gaussian processes	80
J.2	SMOreg	81
	Bibliography	82

List of Figures

2.1	A schematic view of a neural network (image taken from [1]).	13
2.2	An example of a Bayesian network (image taken from [2]).	14
2.3	An example of a decision tree (image taken from [3]).	16
2.4	An example of the results of using a classifier in Weka.	17
2.5	An example of a confusion matrix (image available here [4]).	19
2.6	An example of normal distributions (image taken from [5]).	21
2.7	An example of a set of distributions for 5 input variables (image taken from [6]).	22
2.8	The distributions adapting to the observed data (image taken from [6]).	22
2.9	The distributions adapting to the observed data (image taken from [6]).	23
2.10	An example of mapping input points to a feature space and separating by categories (image taken from [7]).	23
2.11	Graphs depicting different values of the correlation coefficient (image taken from [8]).	24
5.1	The result of using a regression algorithm with the presence of noise. Note the weights being given to the use of the Netscape and Microsoft Internet explorer (MSIE) browsers.	42
A.1	The private repository with the developed system (ask for permission: filipemanuel.lp@gmail.com).	52
B.1	The logic path taken by the server in incoming Init events.	53
B.2	The logic path taken by the library in sending Init events.	54
B.3	The logic path taken by the server in incoming Interval events.	55
C.1	The website used for the questionnaire.	56
C.2	The intro to the questionnaire.	57
C.3	The first step of the questionnaire.	57
C.4	The second step of the questionnaire.	58

C.5	The third step of the questionnaire.	58
C.6	The fourth step of the questionnaire.	59
C.7	The fifth step of the questionnaire.	59
C.8	The sixth step of the questionnaire.	60
C.9	The seventh step of the questionnaire.	60
C.10	The success page of the questionnaire.	61
D.1	Overview of the facebook ads used to gather users.	62
E.1	A screenshot of some of the data gathered.	63
F.1	A screenshot of some of the JavaScript (JS) functions I used to manipulate the gathered data.	64
G.1	A screenshot of the results of using the CVParameterSelection for parameter optimization on the RBFNetwork.	65
G.2	A screenshot of the results of using the RBFNetwork method to classify on age.	66
G.3	A screenshot of the results of using the Bayesian network method to classify on age.	67
G.4	A screenshot of the results of using the CVParameterSelection for parameter optimization on the Rotation forest method.	68
G.5	A screenshot of the results of using the Rotation forest method to classify on age.	69
H.1	A screenshot of the results of using the CVParameterSelection for parameter optimization on the RBFNetwork method.	70
H.2	A screenshot of the results of using the RBFNetwork method to classify on gender.	71
H.3	A screenshot of the results of using the Bayesian network method to classify on gender.	72
H.4	A screenshot of the results of using the CVParameterSelection for parameter optimization on the Rotation forest method.	73
H.5	A screenshot of the results of using the Rotation forest method to classify on gender.	74
I.1	A screenshot of the results of using the CVParameterSelection for parameter optimization on the RBFNetwork method.	75

I.2	A screenshot of the results of using the RBFNetwork method to classify on education.	76
I.3	A screenshot of the results of using the Bayesian network method to classify on education.	77
I.4	A screenshot of the results of using the CVParameterSelection for parameter optimization on the Rotation forest method.	78
I.5	A screenshot of the results of using the Rotation forest method to classify on education.	79
J.1	A screenshot of the results of using Gaussian processes as a regression algorithm on age.	80
J.2	A screenshot of the results of using SMOreg as a regression algorithm on age.	81

Abbreviations

EU European Union. 9

HTTP HyperText Transfer Protocol. 3

JS JavaScript. vi, 3, 7, 27, 37, 64

MSE Mean absolute error. 24, 25

MSIE Microsoft Internet explorer. v, 42

RAE Relative absolute error. 24, 25

RBNetwork Radial basis function network. i–iv, vi, vii, 13, 14, 43–47, 49, 65, 66, 70, 71, 75, 76

RMSE Root mean squared error. 24, 25

RRSE Root relative squared error. 24, 25, 45

SMOreg Sequential minimal optimization for regression. i, iv, vii, 20, 23, 45, 81

SVM Support Vector Machine. 23

US United States. 9

Chapter 1

Introduction

With more than 40% of the world population already online and that number constantly growing[9], businesses which take advantage of internet technologies and social media can develop an important competitive advantage over their peers. This can give them a broader reach, better communication with users and also the possibility of getting more information in a crucial part of developing a successful company - knowledge of customers' demographics and behaviour.

The demographics of the main customers (the target group) help the business owners tailor their service more efficiently, by improving their marketing plan, customer relation and even general branding. Using demographics allow for better business decisions and gives the company a head start in understanding its market.

This information is especially important for small businesses - the knowledge of who their target group is and how it behaves will allow small businesses to adapt themselves to the existing market, sometimes even changing the initial nature of the company.

Nowadays, there are a lot of services trying to provide businesses with this information, ranging from online marketing to consumer relationship solutions or tracking tools.

The abundance of services available, coupled with the effort by businesses to increase knowledge about their customers, show that there exists a market need. The proposal of this thesis is to show there may be a better way of fulfilling it.

1.1 Current Solutions

To get information on the demographics of their users, businesses have several available solutions.

1.1.1 Online Marketing

Some businesses base the knowledge of their target group on online marketing. They reach several user groups they suspect might be their best performing, through marketing channels like Facebook Ads[10] and Google Adwords[11]. From that information they study which ones perform better and bring bigger returns and assume that group to be their target one.

For example, an online shoe shop might create several Facebook ads targeting women from 20-25, 25-30, 30-35, and so on. Then, judging by the sales on the website that they can map to each of those groups, they can say which one seems to be bringing in more revenue.

The downside of this approach is that the company must periodically allocate a budget to test several groups and be able to identify the best performing with some degree of confidence. This budget can sometimes escalate quickly. Moreover, the data is not always reliable as supposedly much of the data from these channels might not be real, but generated by computer programs[12]. Finally, it might not be ideal that this information depends on an external provider, over which the company has no control.

1.1.2 Customer input

Another option is to have direct user input.

With a registration step for example, online businesses can obtain more information about their customers and analyse that data alongside interactions with the product or service with the same goal of finding the best performing demographic.

Intercom[13], an innovative way of communicating with customers, is also another option available. Direct feedback and support can bring a lot of knowledge regarding users.

However, this approach of dealing with sensitive user data may raise data privacy issues and increase the complexity of the existing technological side of the business. In the case of a registration step, it also adds an extra layer of complexity to the sales process that might drive users away and thus have the undesired side effects of driving sales down or hampering sales' growth. Direct communication with the user might also be bothersome and undesired in the user's view.

1.1.3 Online analytics tools

Finally, businesses can also use available online analytics tools to get more information on their users.

The normal usage is by integrating a JS library on their website that stores information regarding a user's behaviour while browsing. Examples of such services are Google Analytics[14] and Mixpanel[15]. The first focuses on general variables of users like time spent on each page, page visits and device information. The second helps to get insights on individual user behaviour throughout the website by knowing the path he took within the website, which buttons triggered him to buy, etc.

This information can then be used to improve the website in order to increase the business' key performance indicators, be them having people browsing longer, increasing revenue per user, increasing brand awareness, and so on.

The biggest downside of this approach is the inability to connect this data with any demographics information (besides location), in order to use them in other aspects of the business besides the website.

1.2 Background

Businesses have always been interested in knowing more about their users and how they use their website. From making better business decisions, to better marketing and online user experience; many are the reasons to get better at knowing who your user is and what his interests are.

Because of that, many studies were made in order to try to understand the user's behaviour while browsing.

Some of these studies focused on analysing the user interaction with the website (usability testing) and how to improve it in order to optimize specific business needs - increasing the checkout funnel conversion, improving specific pages for user retention, and so on.

The papers [16, 17] suggest ways of logging users' browsing data for usability testing purposes, using an HyperText Transfer Protocol (HTTP) proxy. Both papers showed how this information is useful for improving the ease of use of a website and helping create a more enjoyable experience for the user. The authors of [18] also developed client-side intelligent agents to measure, through user behaviour metrics such as scrolling and mouse movement, the interest of a user in a website.

More modern services [19, 20] offered usability testing as a service, providing videos of users utilizing a product in order to get further insights on usage.

Hotjar[21] tracks a number of user behaviour metrics and also records users online, providing this as a service.

Other tracking tools already specified [14, 15] provide different types of usage metrics for the same purpose of optimisation.

Other studies went further and tried to get information on the user, through machine learning algorithms, instead of just studying the actual usage of the website. These studies are mostly using a content-based approach to predict user demographics, using either the content of the websites the user visited or the queries he performed online.

The authors of [22] analysed the possibility of predicting age and gender from browsing behaviour by using content-based and category-based variables in the prediction model. With a dataset of around 189,480 users and 223,786 web pages (from a high traffic website), they showed a correlation between gender/age with the content they read and showed interest on; providing good predictions on gender (79,7% accuracy) and age (60,3% accuracy).

More recently, Google analytics started reporting on users' demographics and interests[23]. Specifically, they provide reports on:

- age
- gender
- affinity categories - identifies users in terms of lifestyle
- in-market segments - identifies users in terms of their product-purchase interests
- other categories - provides a more specific, focused view of users' interests.

These demographics are, as in the previous article, mostly derived from the content/websites the users visit and the apps (s)he uses[24]. The results are also very satisfactory, and provide good predictions on users' demographics and interests[25].

Another attempt to predict the age and gender of online users was made in study [26]. Using words from user queries as input variables, the authors were able to reach 83.8% accuracy on gender prediction.

Paper [27] shows how it was possible to predict demographics from the user's browsing history. Again, using a content-based approach, they were able to reach a

high accuracy of predictions - 80% for age, 76% for gender, 82% for race, 70% for education and 68% for income.

In [28], the authors studied the prediction of gender, age, level of education and profession from online users' clickstream data - the input included URLs of websites visited, number of visits and day of the week and time of the day of the visits.

There are also studies that show to be possible to derive user's demographics by the websites they visit, not related to the content directly. Through known demographics of different websites, like social media networks, one can derive information from which websites the user regularly uses.

For example, in [29], a significant race-dependency preference was found on social networking sites - Hispanic people are more likely to use MySpace than Whites, whom prefer Facebook; while Asians show a preference on other social networks like Xanga and Friendster. Another study [30] found race and education to be related to whether teens chose to use MySpace or Facebook.

Some interesting studies tried going even further and to uniquely identify every user for the purpose of automatic authentication. To help in the problem of the user's online identity the papers [31, 32] developed systems to try and provide a better online experience to users avoiding unnecessary authentication steps. Although the results were not positive, they remain confident on future improved methods.

In a slightly different context, mobile tracking tools have been rising in popularity as the online users are increasingly coming from mobile devices - from last year (2014) even more so than from laptops [33]. To keep up with the usage, most tracking tools have also adapted for mobile tracking and some, like Crashlytics[34] and Flurry[35], even specialized on it. Although not the target of the present thesis, specialising for mobile usage is a must in the future.

As described above, tracking and analysing online user behaviour can be of immense use to businesses and widely studied. By building on the existing research, I will be studying the importance of inputs like mouse movement, click frequency, scrolling speed and so on - input always present while browsing, and not dependent on the particular website - for the same output: predicting user demographics.

1.3 Proposal

From the presented options, the use of analytics tools is in my opinion the most interesting one. It provides an unobtrusive - almost invisible - way of getting crucial information of how people use a website, and, through continuous analysis, how to improve their experience.

The only problem is the lack of information on demographics - it is useful to see how some changes to the website are resulting in terms of time spent by users or revenue obtained, but it would be much better if you could see that a specific gender and age range from a specific country is the one bringing in the most revenue.

The system I propose is similar to the tracking tools presented, with the difference being the importance given in extracting demographics from the data collected through machine learning algorithms. This is useful not only online but in any aspect of a business.

This way, companies would be able to identify their main customers (how old they are, what gender, what is their education, etc) just by analysing the way they browse the website.

The hypothesis I set out to test is then whether there is a correlation between the online browsing behaviour of a user and that user's demographics. Particularly, I plan to gather information regarding the device used to browse - operating system, browser, screen size; and regarding user browsing - number of clicks, scrolling velocity, typing speed, and so on. This information is then used to predict user demographics like gender, age and education.

While the existence of these correlations is doubtful, some of them seem intuitive. Maybe how frequently a user clicks or how fast (s)he scrolls says something about (her)his age - more active if younger, less active if older; or the device information can indicate a social status or profession; or, using the website content, the time spent browsing might reveal something about the user's gender and interests.

The plan is to do the following:

1. Create a tool to gather user data in the browser;
2. Create a machine learning system that can analyse on the gathered data and correlate it with the desired metrics;
3. Collect data from as many sources as possible to improve the system;

4. Analyse the results and application of machine learning algorithms to find correlations;
5. Discuss results and future steps.

The project would start in a similar path to that of analytics services - creating a JS library that would gather all the potentially valuable information on user behaviour.

Afterwards, the creation of the machine learning system would allow us to trace correlations of the browsing behaviour metrics with demographics or the grouping of types of users according to those metrics.

After both systems are done, the biggest obstacle would be gathering the necessary data to feed into the system and allow it to learn. A lot of effort would be put into this to make sure the data is enough, balanced and relevant to the problem in question.

Finally, a thorough analysis of the results and conclusion on exactly what information can be correlated and whether it is worth pursuing or not. Hopefully that will be the time to think about improvements and make it available to the general public.

1.3.1 Integration

Integration with existing businesses would be as simple as any online tracking tool - create an account, import the JS library in the website and that is it.

Data will start to be logged to the server from all kinds of different websites. The machine learning system on the server works as an automated step in the analysis, providing periodical recalculations of predictions to keep businesses up to date.

Businesses could then visit a dashboard with all the information gathered on online usage as well as predictions on the users' demographics.

If successful, this system has the potential to provide an easy, cheap and unobtrusive way of getting information about online users and allow businesses to better understand their target audience and to take better business decisions thereby increasing their future earnings.

1.4 Further possibilities

When applied to a broader variety of websites, such a system would also benefit from knowing the content that each particular website provides or the market in which the particular business resides. This information has already proved to be correlated with the user's age and gender[22].

Another improvement is to start using data regarding specific pages (no. of words, page height, no. of links, etc) to parameterize the system's algorithms - for a page with more links it is normal to have more user clicks; and for different page heights, different scrolling behaviours are expected.

With the usage of mobile devices rapidly increasing[33], specializing in mobile specific statistics is a great add-on to the library.

This service also brings some very interesting future possibilities. With information on target groups for many different businesses, much can be accomplished in terms of marketing and partnerships. For instance, it could recommend users other products or services that the same type of people are using, or suggest partnerships between businesses with the same target group, among others.

Another interesting possibility would be to have client-side code in a website to adapt the design/content of the website to the visitor automatically. This way, several versions of the website could be presented simultaneously to different types of persons, making the browsing experience more enjoyable and user-friendly.

1.5 Limitations

1.5.1 Data and system limitations

When using machine learning for user modelling, big limitations appear: the need for large data sets, the need for labelled data, concept drift and the computational complexity[36].

Although these limitations could be said of any application of machine learning, it becomes a much bigger problem when trying to model people. Because we are such complex systems with erratic behaviour, it is a much more challenging task.

The need for large, labelled datasets becomes even more challenging when considering we are trying to model users through online usage. The amounts of data needed become impossible to gather without the help of a couple of high-traffic websites or through mass emailing.

In machine learning, concept drift means that the properties we are trying to predict can change over time in unforeseen ways, which makes the predictions become less accurate with time[37]. In our case though, the problem is not in the drifting of the variables we are predicting, but on the ones we are tracking and using as input.

The popularity of websites change over time and new ones emerge everyday. The “ruling” web design patterns are constantly changing and will continue so in the future[38]. This makes it so that the behaviour we are studying now will be different in some time - maybe the prevailing design pattern in websites is not the vertical scroll anymore, or people abandon the mouse as the main input tool in a move towards virtual reality. This fact magnifies the problem of concept drift on the inputs. Any system trying to achieve a good modelling of online users must also constantly be evolving to keep up with these changes.

Finally, to be able to manipulate and analyse huge amounts of data, the system has to be of big computationally complexity and thus, error prone and of difficult maintenance.

1.5.2 Ethics

Another limitation that arises when concerning online users’ behaviour is the problem of ethics[39].

Online tracking tools are normally perceived as conflicting with user privacy, more so if people’s personal information is involved. In the European Union (EU), personal data is in fact protected by law; data protection is a right enshrined in the Charter of Fundamental Rights of the EU, whereby its article 8 stipulates that personal data “must be processed fairly for specified purposes and on the basis of the consent of the person concerned or some other legitimate basis laid down by law” [40]. Invading user’s privacy is a major problem and a constant source of issues with authorities (see for instance ruling of 6 October 2015 by the European Court of Justice which essentially invalidated the transfer of personal data collected in the EU to the United States (US) unless “adequate data protection measures” were put in place[41]). These concerns can only be appeased (if at all) with a great effort of transparency to explain why data is being tracked and how that can be useful for the user in question.

While the proposed system is a study out of curiosity and the advantages for businesses is clear, the actual use of it can be seen as unethical. Its practical use should be done very carefully and always keeping an individual user’s data private, while showing only general statistics that can’t be individualized.

Chapter 2

Machine Learning

Put simply, machine learning is the science of getting a computer to act on its own, without being programmed to do so[42]. As a subject of computer science, machine learning has evolved from the study of pattern recognition and computational learning theory in artificial intelligence; its focus being on implementing computer software that can “learn” and act autonomously[43].

In the last decade, machine learning has brought us speech recognition, natural language processing, self-driving cars and even brain-machine interfaces, where a computer learns how to interpret brain waves and can then be controlled with thought alone[42]. It’s also said to be the most promising way of making rapid progress towards true human-like artificial intelligence[44], in the sense that an algorithm that could truly learn on its own could work as (or better than) a human brain.

Machine learning explores the application of algorithms that can “learn” from and make predictions on data. In this case, “learning” is not meant in the “human” sense of the word, but rather a way of finding statistical regularities or other patterns in known data[45]. By finding patterns, the algorithms build a model from the input in order to make predictions on unknown data, being able to make data-driven decisions without human supervision, instead of just following a set of programmed instructions[46].

A computer program is said to learn when it gets better (in a certain measurement) at performing certain tasks the more experience it gets. Or, more formally:

“A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P , if its performance at tasks in T , as measured by P , improves with experience E .” [47]

We will now study how to properly design a learning problem.

2.1 Designing a learning problem

When describing a learning problem we have to define what is the task(T) we want the program to perform, how we'll measure its performance(P) and what experience(E) it has.

For simplicity, imagine we are developing a program that learns how to interpret handwritten words.

For that, we want the program to recognise and classify pictures of written words (T), measure how well it performs by the percentage of words it correctly classifies (P) and, for experience, we provide a database of labelled pictures of written words (E).

Afterwards, we have to consider some attributes that influence how well the program will work, related to the experience it gets.

We have to consider what type of feedback it will receive, whether direct or indirect feedback. In the example, we can give pictures of words and label them correctly (direct) or allowing the program to “read” an entire phrase and then label the phrase as correct or incorrect (indirect).

Obviously, the latter presents a problem of assigning the fault (called credit assignment) - the program knows a phrase is incorrect, but it can't immediately identify the incorrect words.

Furthermore, we should ensure the balance of the dataset; it should have a distribution that represents the unknown data where the algorithm will predict on. Otherwise the results will be biased and not generalizable for unknown data.

Finally, the choice of what algorithms to use can influence the results. There are different types of algorithms that differ in the way they try to represent the dataset. Learning works when algorithms search on a hypotheses space, defined by some representation (linear functions, decision trees, artificial neural networks, and so on) for the hypothesis that best fits the data.

In this way, a neural network algorithm will search for the best neural network representation to fit the dataset, and a decision tree algorithm does the same using decision trees.

We will now study the task of learning and how to measure a system's performance.

Machine learning tasks are normally divided into two main categories regarding ways of learning: supervised vs. unsupervised learning.

2.2 Supervised and unsupervised learning

Supervised learning regards the study of a dataset where the intended output value is known, for later to predict on unseen instances. So an algorithm is first fed a dataset for training where it relates different features to the known output.

Sometimes though, the dataset being study isn't labelled - the intended output is unknown or there is none. The algorithm in this case tries to group the data by checking which ones are similar. These are cases of unsupervised learning.

To better illustrate the difference, I will use a real life example[48]. Imagine you have a basket full of fruit, and your job is to group them by type.

Now, in a first exercise, you learn from a training set that fruits with feature A and B are grapes, and fruits with feature C and D are apples. Because you know this beforehand, when analysing a new fruit, you'll look for the features it has and place it in the correct category - grape or apple. This is supervised learning, because you had a training set with labelled data to help you "learn".

In a second exercise, you don't have any labelled training set, and so you have to group fruits by how similar they look. You can end up with the same results: blue and small fruits to one side and big and red to the other. This happens because you can find similarities between the fruits in each group, like the colour and size. However, the difference here is you don't know they are actually grapes and apples, only that they appear to be two distinct types according to their features. This is called unsupervised learning.

In this thesis, the data to be studied will be labelled, so we will use supervised learning, in particular algorithms for Classification and Regression.

2.3 Classification

Classification is a problem of mapping new, unseen instances to the category they belong; after being trained to do so with labelled training data[49]. This mapping is usually performed when you want to map the inputs to a discrete number of outputs that might represent concepts.

The given example of the fruits is a case of classification: after being given a set of categories and the related features, it is then a job of mapping a new unseen fruit to the right category.

An algorithm that implements classification is called a classifier. In this thesis we'll study the following classifiers: RBFNetwork, Bayesian network and rotation forest.

2.3.1 RBFNetwork

RBFNetwork is an artificial neural network that uses a radial basis function as activation function[50].

Artificial neural networks are the most commonly used type of algorithm for learning[47]. They are a family of statistical learning models with a way of processing information inspired by how nervous systems, like the human brain, process information[51]:

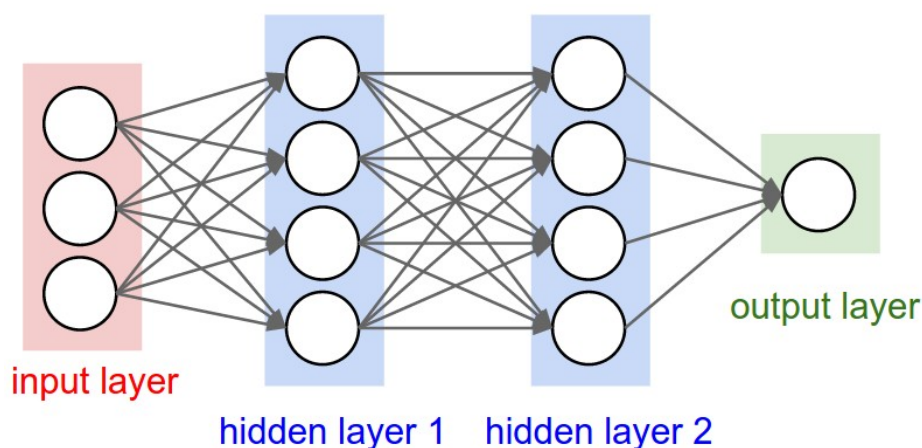


Figure 2.1: A schematic view of a neural network (image taken from [1]).

The motivation for these systems is to capture the complex parallel computation our brains are used to doing easily, by mimicking neurons (here called perceptrons).

As you can see in the picture, the system consists of parallel layers of interconnected nodes working sequentially to solve a specific problem.

Each node processes the inputs with a function (in this case a radial basis function), called the activation function. If the result is over(under) a certain threshold they pass a value of 1(0) to the next layer.

Between each layer, the values are weighted and serve as inputs to the next layer of nodes. These weights can be tuned based on experience, making the system dynamic and capable of learning.

The tuning of these weights is done by an algorithm called Backpropagation. It works by going back from the output propagating the errors (hence the name), and continually tweaking the weights in order to improve the output towards the desired.

Depending on the activation function being used, this model can either be used for classification or regression.

The RBFNetwork uses the k-means clustering algorithms to calculate the best activation function to use (it calculates the centre of the radial basis function). It then learns either a logistic or linear regression according to whether the outputs are to be discrete or continuous, respectively[52].

The algorithm used takes in the following parameters[52]:

- B - Number of clusters (basis functions) to generate. (default = 2).
- S - Random seed to be used by K-means. (default = 1).
- R - Ridge value for the logistic or linear regression.
- M - Maximum number of iterations for the logistic regression. (default -1, until convergence).
- W - Minimum standard deviation for the clusters. (default 0.1).

2.3.2 Bayesian network

A Bayesian network is a probabilistic graphical model that represents relationships among a set of variables of interest via a directed acyclic graph[53].

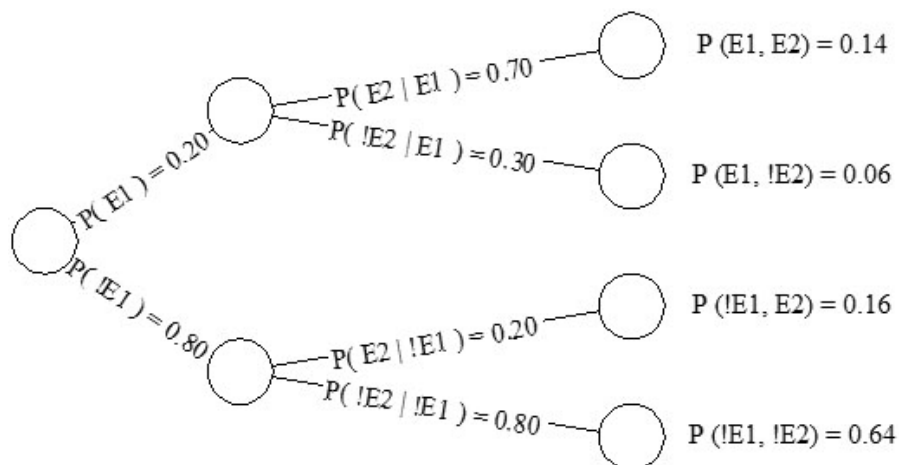


Figure 2.2: An example of a Bayesian network (image taken from [2]).

It represents a complete model of all the variables (represented by the nodes) and the relationships between them (the paths).

Given a new input, a Bayesian network can be used to follow a specific path in the graph and predict the most probable outcome.

Bayesian networks are very important for machine learning because they provide a way of explicitly calculating probabilities for each possible hypothesis.

The algorithm used takes in the following parameters[54]:

- D - Do not use ADTree data structure
- B - BIF file to compare with
- Q - Search algorithm
- E - Estimator algorithm

2.3.3 Rotation forest

Rotation forest is a classifier ensemble method first proposed in 2006 by Rodriguez JJ, Kuncheva LI and Alonso CJ[55].

An ensemble method is a method that uses multiple learning algorithms (normally with a base learner) to obtain better predictions[56]. By combining several algorithms, the ensemble (hopefully) forms better predictions than any of the consistent algorithms on its own.

The rotation forest uses a decision tree for the base learner, as the name suggests.

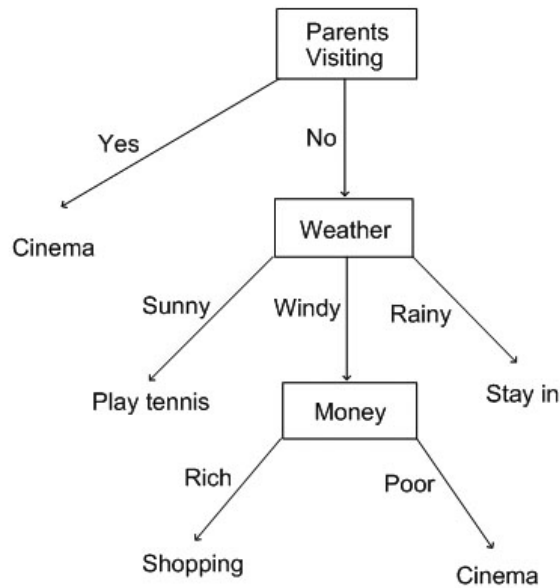


Figure 2.3: An example of a decision tree (image taken from [3]).

Decision tree learning is a method for approximating discrete-valued target functions, using decision trees as the representation.

These algorithms search on a completely expressive hypothesis space (and thus, unrestricted). They are biased to prefer small trees over large trees[47].

The rotation forest ensemble can do classification and regression depending on the base learner. The algorithm used takes in the following parameters[57]:

- N - Whether minGroup (-G) and maxGroup (-H) refer to the number of groups or their size. (default: false)
- G - Minimum size of a group of attributes: if numberOfGroups is true, the minimum number of groups. (default: 3)
- H - Maximum size of a group of attributes: if numberOfGroups is true, the maximum number of groups. (default: 3)
- P - Percentage of instances to be removed. (default: 50)
- F - Full class name of filter to use, followed by filter options.
- S - Random number seed. (default 1)
- I - Number of iterations. (default 10)

- D - If set, classifier is run in debug mode and may output additional info to the console
- W - Full name of base classifier. (default: weka.classifiers.trees.J48)

2.3.4 Analysis

When applying the classifiers to the dataset, the following information will be available for analysis:

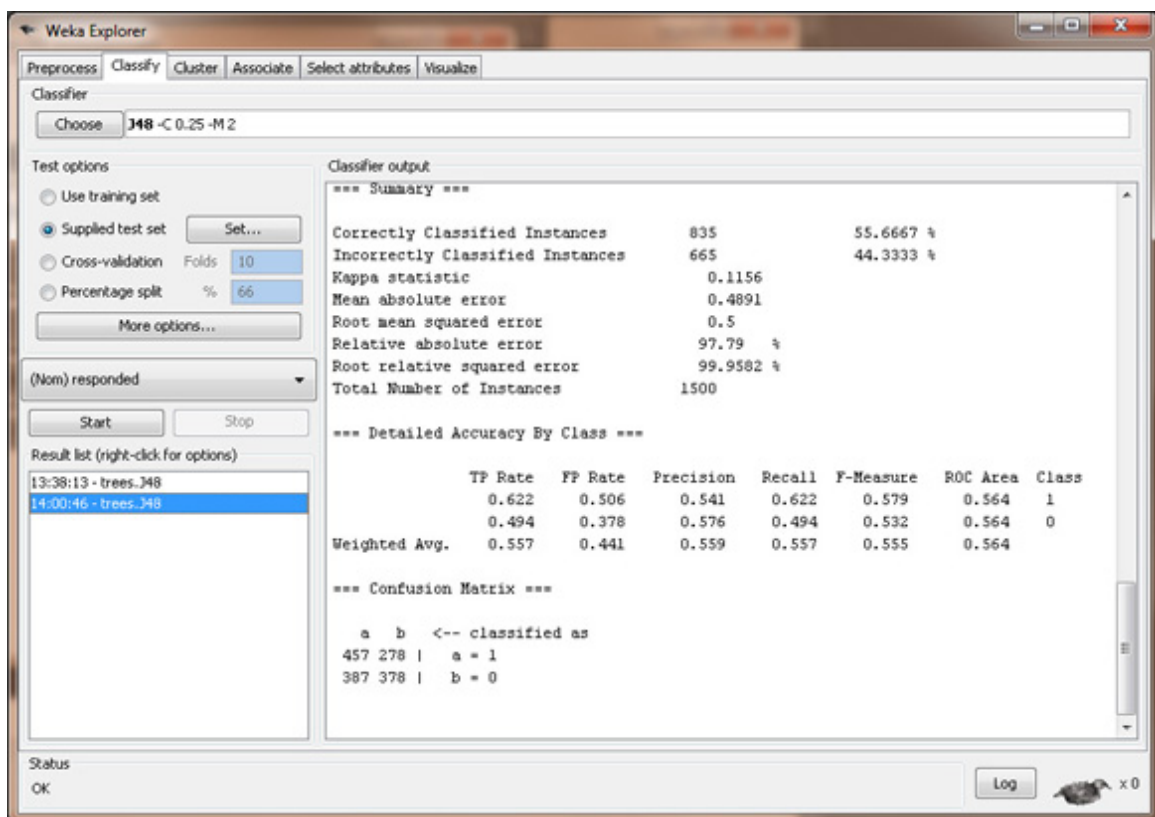


Figure 2.4: An example of the results of using a classifier in Weka.

- Correctly and incorrectly classified instances
- Kappa statistic
- Receiver operating characteristic area (ROC area)
- Confusion matrix

The (in)correctly classified instances tell us how many (and what percentage of the total) of the test instances were (in)correctly classified. The percentage of correctly classified instances is normally called the sample accuracy.

The kappa statistic is a measure of how well the classification agrees with the reality, by correcting for chance[58].

The equation for the kappa statistic is:

$$k = \frac{p_o - p_e}{1 - p_e}$$

Where p_o represents the accuracy of the classifier and p_e the hypothetical probability of chance agreement (the probability of randomly choosing the right categories).

If the kappa statistic is greater than 0 it means the classifier is doing better than by chance, to a maximum of 1, where it has 100% accuracy.

In the “Detailed Accuracy By Class” area, the most important value is the receiver operating characteristic area, or ROC area. This is the area under the ROC curve. The ROC area can be seen as the probability that the classifier will assign a higher score to a randomly chosen positive example than to a randomly chosen negative example[59]. In this way, an optimal classifier would have a ROC area value approaching 1. The value of 0.5 is comparable to random guessing, as the kappa value of 0[60].

The confusion matrix, or error matrix, shows the predicted classes vs. real classes in a matrix with how many were correctly or incorrectly placed in each category[61].

		prediction outcome		total
		p	n	
actual value	p'	True Positive	False Negative	P'
	n'	False Positive	True Negative	N'
total		P	N	

Figure 2.5: An example of a confusion matrix (image available here [4]).

With this matrix you can get a better feel for what is happening and if the system is confusing any categories.

Finally, to compare different models in terms of performance, we can calculate the expected error of each and compare them.

First, we get the error on the analysed dataset, called the sample error, by dividing the number of incorrectly categorized instances by the number of total instances.

The real error is the same but calculated over the entire unknown distribution of examples, where the algorithm is supposed to act. It represents the probability of the algorithm to incorrectly categorize an unknown instance.

Because this error is impossible to calculate, we can approximate and get an interval where it is to a certain confidence. To calculate the range where the true error might be present with 95% confidence we use the following formula:

$$error_s(h) \pm 1.96 \sqrt{\frac{error_s(h)(1 - error_s(h))}{n}}$$

Where $error_s(h)$ is the error that hypothesis h has in sample S , and n is the number of instances tested. For an interval with other degree of confidence, we just have to replace the factor 1.96 (99% - 2.58, 90% - 1.64, 80% - 1.28, 68% - 1).

Without getting into much detail, this formula is derived by assuming the distribution of the errors present in several samples follows a Binomial distribution. The binomial distribution can also be approximated, for a large number of instances, by a Normal distribution; which has been extensively studied and is easier to deal with.

Lastly, to compare two different hypothesis (or models), we can approximate what the differences on their true errors would be. We start by calculating the difference of the sample errors:

$$d_s = error_s(h_1) - error_s(h_2)$$

Where d_s is the difference of errors in the studied sample and h_1 and h_2 are the two hypotheses used.

We can then get an interval with 95% confidence:

$$d_s \pm 1.96 \sqrt{\frac{error_s(h_1)(1 - error_s(h_1))}{n} + \frac{error_s(h_2)(1 - error_s(h_2))}{n}}$$

As previously, for a different degree of confidence, we can replace 1.96 for another factor.

This difference will then allow us to compare the two models and draw our conclusion on which is the best performing.

2.4 Regression

Regression is a statistical process to estimate relationships between variables[62]. Given a number of variables as input, the job is to find a mapping between those variables and the known output.

This mapping is used when you want to map the inputs to a continuous number of outputs, as opposed to discrete values in Classification.

An example is how you might try to map the numbers “1,2,3,4,5,..” to “1,4,9,16,25,..” in order to find a relation between them: the latter being the second power of the former.

In this thesis I will use Gaussian processes and SMOreg as regression algorithms.

2.4.1 Gaussian processes

Gaussian processes are a family of statistical distributions said to be a generalization of the Gaussian probability distribution. In a Gaussian process, every input variable is associated with a normally distributed random variable, and every collection of the variables has a multivariate normal distribution[63].

A normal distribution (Gaussian distribution, or even bell curve) is a very common probability distribution that is often used to represent random variables whose distributions are unknown in many different fields, from natural to social sciences[5].

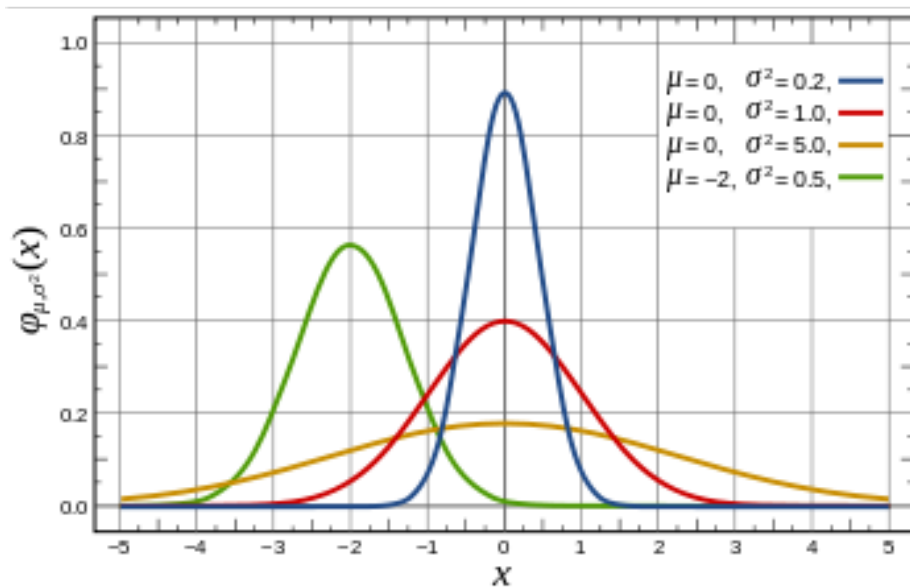


Figure 2.6: An example of normal distributions (image taken from [5]).

When used for regression, each input variable has its own normal distribution, a prediction of the data:

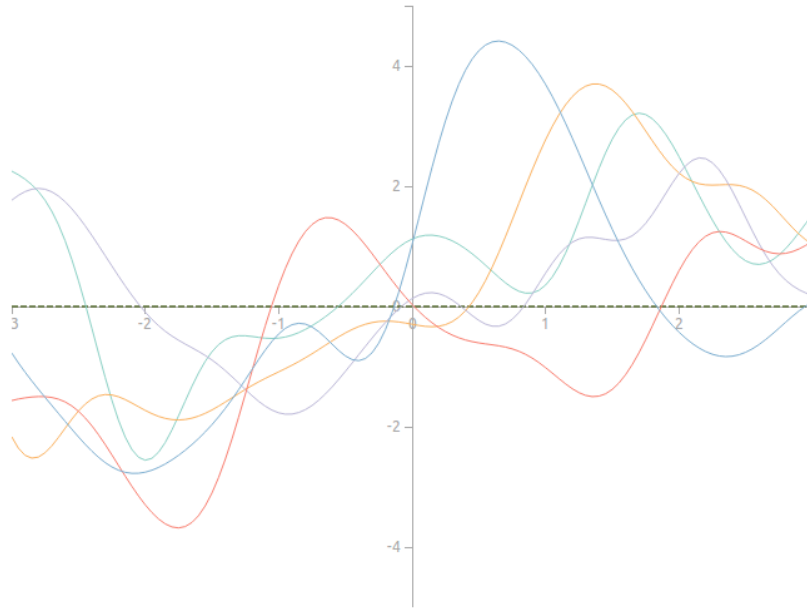


Figure 2.7: An example of a set of distributions for 5 input variables (image taken from [6]).

As the data is inputted, the system creates a distribution (using weights of the others) that adapts to the observed data gradually:

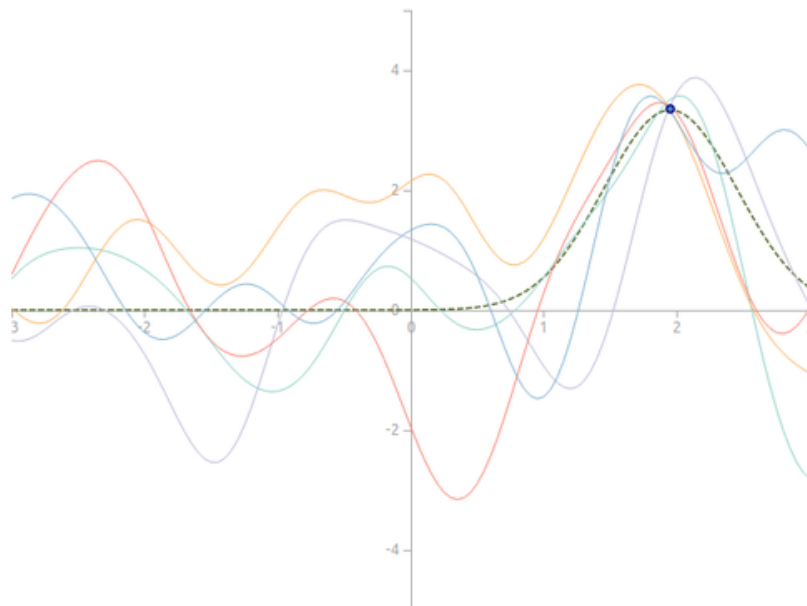


Figure 2.8: The distributions adapting to the observed data (image taken from [6]).

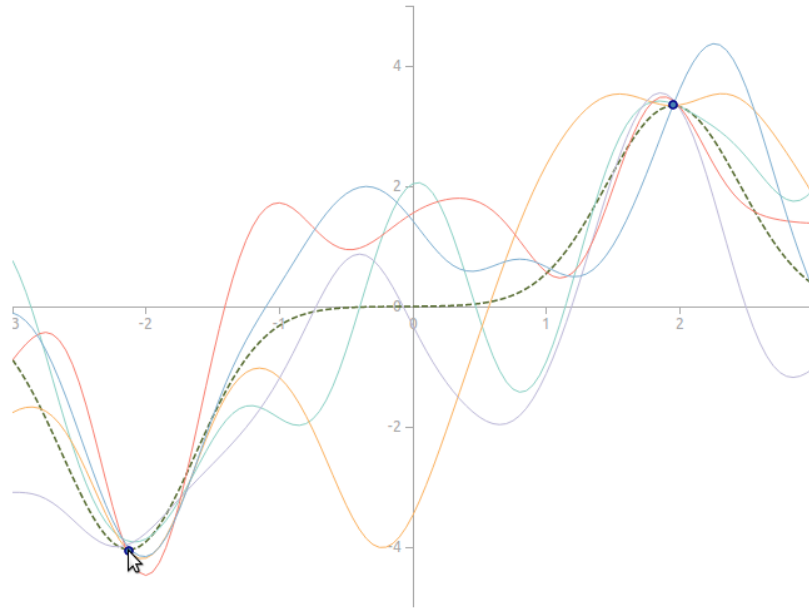


Figure 2.9: The distributions adapting to the observed data (image taken from [6]).

The result is then a smooth function that models the given data and infers the output of unknown values.

2.4.2 SMOreg

The SMOreg is an algorithm that implements the Support Vector Machine (SVM) for regression[64].

An SVM is supervised learning model that can be used for both regression and classification[65]. It works by representing all input values as points in a plane mapped by their features, as shown in the following picture:

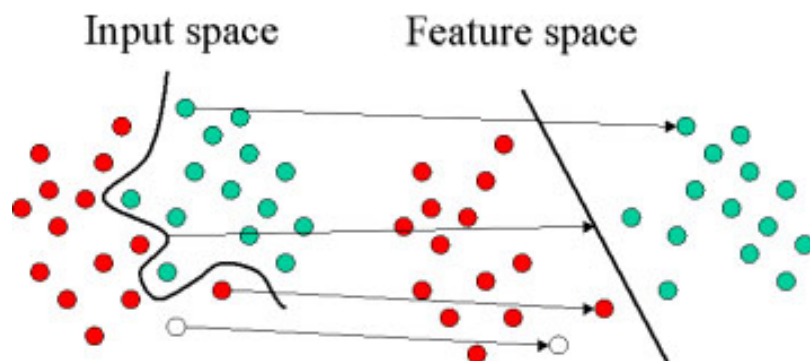


Figure 2.10: An example of mapping input points to a feature space and separating by categories (image taken from [7]).

After this mapping, it separates the points into categories, known as kernels, with gaps between them as wide as possible. The line separating the kernels is called a decision boundary.

To predict a new point, it maps the point the same way and sees on which side of the decision boundary it sits.

In a case with more features, the algorithm constructs hyperplanes as decision boundaries (instead of lines) in a multidimensional space (instead of two dimensional).

2.4.3 Analysis

When applying the regression algorithms to the dataset I will be able to analyse two pieces of information: the correlation coefficient and the errors.

The correlation coefficient represents how well the outputs correlate or depend on the inputs.

A correlation coefficient of 0 means the variables are not correlated at all. As it approaches -1 or 1 it means a better and better negative correlation (when one increases, the other decreases) or positive correlation (either increase or decrease with each other) respectively.

The following picture shows different values of correlation for better visualization.

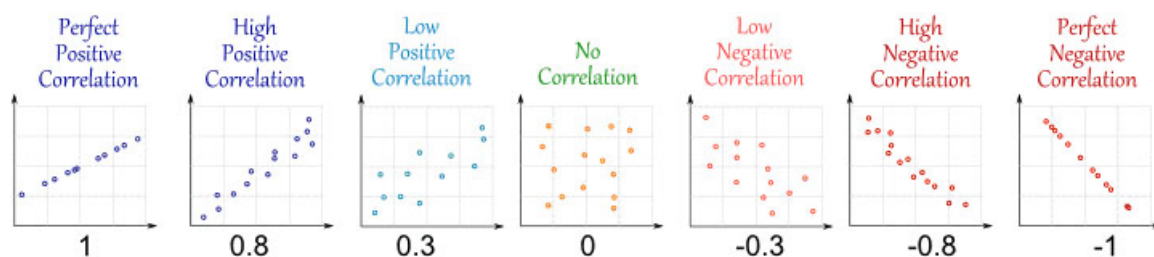


Figure 2.11: Graphs depicting different values of the correlation coefficient (image taken from [8]).

The errors we get are the Mean absolute error (MSE), Root mean squared error (RMSE), Relative absolute error (RAE) and Root relative squared error (RRSE):

$$MSE = \frac{1}{N} \sum_{i=1}^N |\hat{\theta}_i - \theta_i|$$

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (\hat{\theta}_i - \theta_i)^2}$$

$$RAE = \frac{\sum_{i=1}^N |\hat{\theta}_i - \theta_i|}{\sum_{i=1}^N |\bar{\theta} - \theta_i|}$$

$$RRSE = \sqrt{\frac{\sum_{i=1}^N (\hat{\theta}_i - \theta_i)^2}{\sum_{i=1}^N (\bar{\theta} - \theta_i)^2}}$$

These errors are different ways of comparing the true values with the estimates of the model.

The MSE and RMSE are calculated using the average difference between each true value - estimate pair and summing them. The way the RMSE is calculated makes it extra sensitive for estimates further from the true values.

The relative errors are the absolute errors divided by the scale of the true values and multiplied by 100 to get a percentage.

2.5 Cross-validation

For validating how well a particular algorithm's predictions generalize, I will be using cross-validation.

Cross-validation is a model validation technique that studies how well the predictive model generalizes, in order to give accurate predictions on unknown, independent data[66].

It works by partitioning the data into n complementary sets. It then performs the analysis on each of the possible n-1 sets, and uses the last to validate, as a testing set. The final error is the average of the errors resulting from each of the n analyses.

2.6 Weka

In the present thesis, for the analysis of the data, I will be using Weka.

Weka is a collection of machine learning algorithms for tasks regarding data mining and analysis[67]. The Windows program can load a dataset and analyse it using several algorithms both for supervised and unsupervised learning. It can also be

integrated in another system using their JAVA library or its adaptations to other programming languages.

A big aspect of using Weka is that it offers ways of automating the process of tuning the optimal parameters to use with an algorithm, which is a very tedious process when done manually.

Choosing the dataset to test and the pretended algorithm, it allows you to choose a range where you want the parameter to be tested, and it finds the optimal value.

From the three possible algorithms for parameter optimization, I will be using `CVParameterSelection`.

`CVParameterSelection` performs the selection of the parameters for any classifier by cross-validation. It can optimize an arbitrary number of parameters, but it cannot optimize on nested parameters - parameters not fed directly to the algorithm in question.

Chapter 3

Implementation

To develop the proposed system I needed to develop a library in JS to be integrated in a website, a server side app to handle the requests with user information and a database to store the data. Besides this, I needed access to a website to test the system and gather data.

The machine learning algorithms were not automated in the system, but tested manually with the data set gathered, for better visualization and analysis.

The JS library was written in JQuery[68], the server side app was written in node.js[69] and the database used was MongoDB[70]. The entire app is available for consulting in a private repository on Github[71] (see appendix A).

The app was hosted on Heroku[72] to test in a real website.

I will now explain the different parts of the system and how this can be integrated in an existing business.

3.1 Database models

To store the data logged I divided it into different concepts: interval, page, session and device.

An **Interval** represents a unit of time with information on the user's behaviour. It can be of two types, a default and an unload interval. The default interval is an entity with information on how a user behaved in the website every 5 seconds; the unload has information on how a user behaved in the website moments before the user closes it (the unload event of the window). The unload interval is not guaranteed to always exist, we can never be sure that request gets sent before the page is actually closed.

These intervals can then be grouped to create page statistics. Each **Page** represents a page view and, with the information of each interval, can have data of how the user behaved in that particular page.

In the same way, pages can be grouped into Sessions and sessions into Devices. A **Session** represents a normal user session - the user navigation through the website. The **Device** represents the entire browsing data of the user.

The inability to distinguish a user across devices makes it impossible to group device information on a user entity. This might be achieved in other ways in the future.

These are the database models of the specified entities.

3.1.1 Device Model

Uniquely identifies the currently connected device.

- `browser` (string)
- `browserVersion` (string)
- `isMobileOrTablet` (boolean)
- `operatingSystem` (string)
- `screenWidth` (number)
- `screenHeight` (number)
- `created_at` (date)
- `updated_at` (date)

3.1.2 Session Model

A session is defined as a normal user session - a continuous set of page views not very spaced out in time.

- `device` - reference to a device object.
- `created_at` (date)
- `updated_at` (date)

3.1.3 Page Model

Defines a page view and information regarding the page visited.

- `pathname` (string) - The path of the current page.
- `documentWidth` (number).
- `documentHeight` (number).
- `viewportWidth` (number).
- `viewportHeight` (number).
- `session` - reference to a session object.
- `created_at` (date)
- `updated_at` (date)

3.1.4 Interval Model

Stores user browsing behaviour. Each 5 seconds, a DEFAULT interval is created, and on unloading of the window a UNLOAD interval is created. All properties are relative to that interval only and are reset between intervals.

- `type` (string) - type of interval; DEFAULT if normal periodic interval, UNLOAD if window unload interval.
- `userIsActive` (boolean) - if user has been active in that interval.
- `nrOfClicks` (number) - total no of clicks.
- `nrOfClicksOnClickables` (number) - no of clicks on links and buttons.
- `nrOfCharactersTyped` (number)
- `scrollPositions` (array)
- `mousePositions` (array)
- `page` - reference to a page object.
- `created_at` (date)
- `updated_at` (date)

3.2 Software

3.2.1 Events

The frontend library sends data to the server in two different events: the Init event and the Interval event.

The init event will be triggered every time a page in the website is opened. This can happen in 3 different cases:

- The first time the user visits a website,
- The user starts a new session,

- The user opens a new page while navigating normally.

The first time the user visits a website, the init event will create a new **Device** to identify the user. The server responds with the created device id that is stored in the user's computer in a cookie. All subsequent requests send this id so the user can be identified.

In the case that it is not the first time visiting the website, the init event will create either a new **Session**, if the user hasn't been active for a while; or a **Page** view, if the user is navigating the website normally.

The interval event is triggered in fixed time periods (each 5 seconds) or at window unloading, and sends browsing behaviour information along with the device id. This is used to keep track of the user's browsing data and associate it to the correct device.

3.2.2 Information tracked

There are several types of information being gathered: device information, page information and user browsing information.

3.2.2.1 Device information

Device information is information regarding the computer/mobile and browser the user is using. This information is the initial data that gets sent to the server:

- Browser
- Browser version
- If mobile/tablet or desktop
- Operating System
- Screen width
- Screen height

3.2.2.2 Page information

Page information is data regarding the specific page that is being visited, like path-name, page height, and so on.

- Pathname
- Document width
- Document height
- Viewport width
- Viewport height

3.2.2.3 User browsing information

User browsing information is data regarding the browsing of the user. This information is sent to the server every 5 seconds.

- If user is active
- Number of clicks
- Number of clicks on links or buttons
- Number of characters typed
- Scroll positions
- Mouse positions

3.2.3 Server side app

The server side app takes care of handling incoming data of the events specified. It gets the information and stores it according to the different possible user scenarios.

3.2.3.1 Init event handling

When the server receives the init event, the following happens.

- If there isn't a device id specified or that id doesn't exist, the server creates a **Device**, **Session** and **Page**.
- If device id exists, server checks if the session is valid (see 3.2.3.3).
- If session is not valid, a new **Session** and **Page** are created.
- If session is valid, a **Page** is created.
- In any of the outcomes, server sends back the device id to the frontend.

The diagram in appendix B, page 53 shows the logic path taken by the server in incoming init events.

3.2.3.2 Interval event handling

This is how the server handles interval events:

- With device id, server gets device, last session and last page of the user.
- Server checks if session is valid (see 3.2.3.3).
- If session is not valid, a new **Session**, **Page** and **Interval** are created.
- If session is valid, an interval is created.

The diagram in appendix B, page 53 shows the logic path taken by the server in incoming interval events.

3.2.3.3 Session validity definition

A session is supposed to mimic a normal user session in a website - the user can navigate through different pages, can keep one open while away from the computer, but it will only be considered ended when no more intervals are being sent (either closing the website, no internet, computer turning off, etc).

In the server, a session is said to be valid if the last interval for the specified device is recent (less than 30 seconds ago). If the last interval is older than that, session is considered invalid and therefore a new one is created.

3.2.4 Frontend library

The frontend library takes care of gathering data regarding user browsing and sending it to the server for storage.

3.2.4.1 Init event handling

- The library reads the device id from a cookie.
- If the device id doesn't exist, it sends an init event with page information (3.2.2.2) plus device information (3.2.2.1).
- If the device id exists, the library sends an init event with device id and page information (3.2.2.2).

The diagram in appendix B, page 53 shows how the library sends the init request.

3.2.4.2 Init interval handling

Every 5 seconds the library is gathering information on the user browsing (3.2.2.3). The interval then gathers the data and sends it to the server, resetting it for the subsequent intervals.

When the user closes the window, the library tries to send the information of the last moments for consistency. This information is not guaranteed to get to the server before the page closes.

3.2.5 Machine learning system

The machine learning system was not automated into the system in the present thesis. Instead, the gathered data was tested manually using Weka for a better visualisation and analysis.

The integration of the best performing algorithms in the system would be, however, the next logical step.

Chapter 4

Data gathering and manipulation

The next step in order to study the proposed hypothesis is the gathering and manipulation of data.

For any machine learning algorithm this step is crucial - it is decisive to the accuracy of the algorithm.

4.1 How the data was gathered

Starting with a product I am developing in parallel (www.stellared.com, an education startup not related with the present thesis) and with the purpose of getting browsing information, I added a questionnaire to the website.

This questionnaire was designed in a way that would make people use the website for a while. It included radio buttons, multiple choice and text fields so it could get a range of different user behaviour logged.

The questions were focused on getting information I would use in the development of the product in question but also with requests for gender, age and education of the user. This way I can label the gathered user behaviour data.

In appendix C (page 56) you can see the screenshots of the website and the questionnaire steps.

4.2 How the users were obtained

To get a balanced representation of the entire population (i.e. unbiased towards any specific demographic), I started by using Facebook advertisement.

I created a campaign on Facebook with ad sets targeting different groups: people with age between 15-25, 26-35 and 36-50; male and female; from Portugal, Germany and the United States. The 18 created ads ran for a total of 3 days.

In appendix D, page 62 is a screenshot of the ads used.

Unfortunately, the result of the ads was not satisfactory, and, after some expense, continuing this path was not feasible.

After failing to get enough users to the website through online marketing, I asked the Director of Nova University to send out an e-mail to all students while I asked friends and family to help by filling out the questionnaire in the website.

By doing so, the dataset of users may have been somewhat biased towards young people; but this allowed me to obtain data from the 145 replies to the questionnaire and a total of 524 visitors on the course of 5 days.

4.3 Data gathered

The data gathered consisted of a total of 524 users, 832 sessions, 929 page views and 75,823 intervals.

From those, a total of 145 replied to the questionnaire and I was able to label the information of these users with their demographics - more specifically, their age, gender and education.

Some demographics of the users:

Age

- 17 or younger - 2 (1.38%)
- 18 to 21 - 16 (11,03%)
- 22 to 25 - 45 (31,03%)
- 26 to 29 - 25 (17,24%)
- 30 to 39 - 44 (30.34%)
- 40 or older - 13 (8,97%)

Gender

- Male - 86 (59,31%)
- Female - 59 (40,69%)

Education

- High-school level or lower - 15 (10,34%)
- Bachelor's Degree - 59 (40,69%)
- Master's Degree - 67 (46,21%)
- PhD or higher - 4 (2,76%)

A screenshot of the data gathered can be viewed in the appendix E, page 63.

The data collected is not enough for a thorough analysis of the hypothesis put forward in this thesis. Nevertheless, a small correlation can show up with a small amount of data and that could be a good sign, hoping it's not purely coincidental.

In the worst case scenario, the process is now documented so it can be pursued in the future in better conditions and with more data.

4.4 Data manipulation

The data gathered is not yet ready to be used in the machine learning systems.

With the data from the users which filled in the questionnaire, I manipulated it to get more meaningful information regarding the user browsing behaviour.

The information for each interval was grouped to create statistics for each page visit, the pages were grouped to create session statistics, and the sessions were grouped to create device/user statistics.

This way we can have the user statistics we need for the study.

To calculate some of these variables, mainly regarding mouse and scrolling positions, I wrote a set of JS scripts that you can see in appendix F. These functions looped through the values of scrolling positions (and mouse positions) to calculate distance scrolled(moved) and average velocity of scrolling(moving).

The variables that showed up in the manipulation of the data were: typingSpeed, frequency of clicks, time active/inactive, distance scrolled and average scrolling speed, distance moved with the mouse and average speed of that movement.

Afterward I organized the variables and removed useless data, like tracking errors arising from the use of uncommon mobile phones or browsers.

Finally, I stored the data in a .csv file ready to be used in Weka.

In the following chapter I will study if this information on users' activity (frequency of clicks, scrolling/moving speed, etc) and their device information (operating system, browser, screen sizes, etc) give any insights to the users' gender, age or education.

Chapter 5

Data analysis and Discussion

I will start by defining the learning problem at hand:

Task: predict user demographics from user online behaviour

Performance measure: percentage of correctly classified users

Experience: database of user browsing variables with given classification

In this task, the data will be given in a direct feedback way. Furthermore, because it is labelled, I will be using supervised learning for the analysis.

To check the balance of the distribution of data against that of the entire distribution, I compared the values of operating system and browser I had with the global usage of operating systems[73](approximated) and browsers[74]. The rest of the input features are not possible to compare as they might be different in an individual level.

- Windows - 61.5% (dataset), ~88% (real)
- Mac OS - 23.1% (dataset), ~7.72% (real)
- Linux - 14.7% (dataset), ~1.74% (real)
- Chrome - 76.9% (dataset), 41.4% (real)
- Firefox - 10.5% (dataset), 10.4% (real)
- Safari - 8.39% (dataset), 20.4% (real)

We can see the dataset is a bit unbalanced regarding operating system - on the usage of Windows (too little) and Mac OS (too high). Linux is not possible to be compared as the ones in the dataset also include Android phones, and the real value does not.

As for browsers, we see a slight bias towards the usage of Chrome. The usage of Safari is also explained since the dataset does not include any users with iPhones, while the real value includes them.

Likewise, it is clear that a distribution problem on the output is present - over 61% of users are either between 22 and 25 or 30 to 39 years old, and almost 87% has either a Bachelor or Master's degree.

The not-so-high number of instances (n=143) and the over-representation of particular subsets show that the dataset is not big enough or well distributed.

This may suggest that the analysis might be coincidental to a certain extent; and that a larger dataset would be desirable in order to derive better results.

An unbalanced dataset might make an algorithm seem good while it is actually only assigning most of the instances to one or two categories, thus not really generalizable.

The variables I will be using as inputs are:

- browser
- operating system
- screen height
- screen width
- if is mobile
- clicks per active interval
- clicks on links/buttons per active interval
- characters typed per active interval
- time on page
- time on page active

- range scrolled
- distance scrolled
- average scrolling speed
- distance moved
- average moving speed

Although these are already representative of a range of possible user behaviours, the question always remains if these are sufficient to represent a browsing session or if there are other crucial variables that were overlooked.

The user demographics I am studying are age, gender and education. Because these demographics are separated into several ranges, I will use classification to map the features into each category.

Afterwards, and by associating a number to each age range, I will also study the relation between the inputs and age through regression.

Before applying each algorithm, I will be using `CVParameterSelection` to optimize the parameters to use with each algorithm.

To better validate the analysis, I will use cross-validation by dividing the dataset onto 10 subsets.

5.1 The presence of noise

After a preliminary analysis of the algorithms I found that some features were weighted too much although they were not generalisable.

There are only two instances of users with Netscape or Internet Explorer as browsers, but the use of these browsers was very influential on the algorithms:

```
Classifier output

=== Classifier model (full training set) ===

SMOreg

weights (not support vectors):
+ 0.0853 * (normalized) browser=Firefox
+ 0.1509 * (normalized) browser=Chrome
- 0.1358 * (normalized) browser=Safari
+ 0.4342 * (normalized) browser=MSIE
- 0.5345 * (normalized) browser=Netscape
+ 0.2032 * (normalized) isMobileOrTablet
- 0.0004 * (normalized) operatingSystem=Windows
+ 0.0872 * (normalized) operatingSystem=MacOS
- 0.2519 * (normalized) operatingSystem=Linux
+ 0.1651 * (normalized) operatingSystem=UNIX
+ 0.5342 * (normalized) screenHeight
- 0.1668 * (normalized) screenWidth
- 0.264 * (normalized) clicksPerIntervalActive
- 0.0336 * (normalized) clicksOnClickablesPerIntervalActive
+ 0.2094 * (normalized) charactersTypePerIntervalActive
- 0.7059 * (normalized) timeOnPage
+ 0.2103 * (normalized) timeActive
- 0.2381 * (normalized) rangeScrolled
+ 0.1974 * (normalized) distanceScrolled
- 0.2016 * (normalized) averageScrollingSpeed
+ 0.0894 * (normalized) distanceMoved
- 0.0771 * (normalized) averageMovingSpeed
+ 0.3825
```

Figure 5.1: The result of using a regression algorithm with the presence of noise. Note the weights being given to the use of the Netscape and MSIE browsers.

After noting this, I removed these instances, ending up with 143 instances for analysis.

The removal of the noise made the accuracy of the classifiers and the correlation coefficient worst, showing how much the presence of noise in the training data influences the algorithms' success.

5.2 Age

The age was separated into six categories; "17 or younger", "18-21", "22-25", "26-29", "30-39" and "40 or older".

5.2.1 Classification

5.2.1.1 RBFNetwork

Using CVPParameterSelection (see image G.1) to optimize the parameter B(number of clusters) from 2 to 5, I found out the best performing is B=3.

The RBFNetwork (with parameters “-B 3 -S 1 -R 1.0E-8 -M -1 -W 0.1”, see image G.2), correctly classified ~31.57% of the instances with a kappa statistic of 0.0792.

The error on this dataset is 68.53% and the expected error is in the interval (with 95% confidence):

$$60.92\% < error < 76.14\%$$

5.2.1.2 Bayesian network

For the Bayesian network, the default parameters were used. The parameters of the search and estimator algorithms, used within the Bayesian network, could not be tuned due to the limitation of the optimization algorithm (does not optimize nested parameters).

The Bayesian network (with parameters “-D -Q weka.classifiers.bayes.net.search.-local.K2 -P 1 -S BAYES -E weka.classifiers.bayes.net.estimate”, see image G.3), had an accuracy of ~30.77% and kappa statistic of 0.0291.

The error on this dataset is 69.23% and the expected error is in the interval (with 95% confidence):

$$61.67\% < error < 76.79\%$$

5.2.1.3 Rotation Forest

Finally, for the rotation forest, the CVPParameterSelection (image G.4) optimized the minimum(G) and maximum(H) number of groups in the ranges 1 to 3 and 3 to 5 respectively. The optimal parameters are G = 2 and H = 5.

The rotation forest (with parameters “-N -G 2 -H 5 -P 50 -F weka.filters.unsupervised.-attribute.PrincipalComponents -R 1.0 -A 5 -M -1 -S 1 -I 10 -W weka.classifiers.trees.J48 -C 0.25 -M 2”, see image G.5), had an accuracy of ~32.87% and kappa statistic of 0.1026.

The error on this dataset is 67.13% and the expected error is in the interval (with 95% confidence):

$$59.43\% < error < 74.83\%$$

5.2.1.4 Analysis

The results for classification in terms of the age are not very satisfactory.

All algorithms reached around 30%-33% of correct predictions. The positive kappa statistic shows it's better than chance, but with very little improvement.

In the confusion matrix we can also see that the categories of "22 to 25" and "30 to 39" are confused by the algorithm and also over represented. That might lead to a higher accuracy that is not really generalizable.

By calculating the real difference (d) in errors between the algorithms, we reached the following intervals, with the given degree of confidence.

RBFNetwork - Bayesian Network

$$-0.114314 < d < 0.100314(95\%)$$

$$-0.0770827 < d < 0.0630827(80\%)$$

RBFNetwork - RotationForest

$$-0.0942657 < d < 0.122266(95\%)$$

$$-0.0567041 < d < 0.0847041(80\%)$$

Bayesian Network - RotationForest

$$-0.0869372 < d < 0.128937(95\%)$$

$$-0.0494896 < d < 0.0914896(80\%)$$

From the analysis of the differences in accuracy we cannot say with a good degree of confidence which algorithm was the best performing, even though the Rotation forest had a better accuracy in this dataset.

5.2.2 Regression

Because age is something continuous, I tried to assign a number to each of the ranges in order to try regression algorithms. This way, “17 or younger” is now represented as 1, “18 to 21” as 2, and so on.

This conversion is very error prone since the several intervals do not have the same range (some have only 3 years, others with more than 10), but it still might be worth to test it.

In appendix J, you can see the results of the regression algorithms on the age.

The use of the Gaussian processes algorithm held a correlation coefficient of 0.1591 with a RRSE of 100.06% (J.1).

Using SMOreg the correlation coefficient was 0.1619 and the RRSE 107.03% (J.2).

This analysis found no significant correlation between the inputs and output.

To make the results more conclusive, we’d need a bigger amount of instances to train the algorithm, a more balanced dataset and a more correct mapping of categories to numerical values or considering the age a continuous variable since the beginning, including during the gathering of data.

5.3 Gender

The gender was separated into two categories; “Male” and “Female”.

5.3.1 RBFNetwork

Using CVParameterSelection (see image H.1) to optimize the parameter B(number of clusters) from 2 to 5, I found out the best performing is B=3.

The RBFNetwork (with parameters “-B 3 -S 1 -R 1.0E-8 -M -1 -W 0.1”, see image H.2), correctly classified ~58.04% of the instances with a kappa statistic of 0.0926.

The error on this dataset is 41.96% and the expected error is in the interval (with 95% confidence):

$$33.87\% < error < 50.05\%$$

5.3.2 Bayesian network

For the Bayesian network, and for the same reason, I used the default parameters.

The Bayesian network (with parameters “-D -Q weka.classifiers.bayes.net.search.-local.K2 -P 1 -S BAYES -E weka.classifiers.bayes.net.estimate.SimpleEstimator -A 0.5”, see image H.3), had an accuracy of ~64.34% and kappa statistic of 0.1702.

The error on this dataset is 35.66% and the expected error is in the interval (with 95% confidence):

$$27.81\% < error < 43.52\%$$

5.3.3 Rotation Forest

Finally, for the rotation forest, the CVPParameterSelection (image H.4) optimized the minimum and maximum number of groups (G and H) in the ranges 1 to 3 and 3 to 5 respectively. The optimal parameters are $G = 3$ and $H = 5$.

The rotation forest (with parameters “-N -G 3 -H 5 -P 50 -F weka.filters.unsupervised.attribute.PrincipalComponents -R 1.0 -A 5 -M -1 -S 1 -I 10 -W weka.classifiers.trees.J48 -C 0.25 -M 2”, see image H.5), had an accuracy of ~60.14% and kappa statistic of 0.1022.

The error on this dataset is 39.86% and the expected error is in the interval (with 95% confidence):

$$31.84\% < error < 47.89\%$$

5.3.4 Analysis

When classifying in terms of gender the results were better, but still not very promising.

The algorithms had an accuracy between 58% and 64%. The kappa statistics were positive, around 0.1-0.17, thus little better than by chance.

By calculating the real difference (d) in errors between the algorithms, we reached the following intervals, with the given degree of confidence.

RBFNetwork - Bayesian network

$$-0.0497854 < d < 0.175659(95\%)$$

$$-0.0106776 < d < 0.136552(80\%)$$

$$0.00542559 < d < 0.120448(68\%)$$

RBFNetwork - RotationForest

$$-0.0929605 < d < 0.134919(95\%)$$

$$-0.0534305 < d < 0.0953885(80\%)$$

Bayesian network - RotationForest

$$-0.154225 < d < 0.0703089(95\%)$$

$$-0.115275 < d < 0.0313592(80\%)$$

We can not again distinguish the best performing algorithm with a good degree of confidence.

From the higher accuracy present in the Bayesian network, we can say it performed better than the RBFNetwork at 68% confidence.

5.4 Education

Finally, education was separated into four categories; “High-school level or lower”, “Bachelor’s degree”, “Master’s degree” and “PhD or higher”.

5.4.1 RBFNetwork

Using CVParameterSelection (see image I.1) to optimize the parameter B(number of clusters) from 2 to 5, I found out the best performing is B=3.

The RBFNetwork (with parameters “-B 3 -S 1 -R 1.0E-8 -M -1 -W 0.1”, see image I.2), correctly classified ~46.15% of the instances with a kappa statistic of 0.0951.

The error on this dataset is 53.85% and the expected error is in the interval (with 95% confidence):

$$45.68\% < error < 62.02\%$$

5.4.2 Bayesian network

For the Bayesian network I used the default parameters.

The Bayesian network (with parameters “-D -Q weka.classifiers.bayes.net.search.-local.K2 -P 1 -S BAYES -E weka.classifiers.bayes.net.estimate.SimpleEstimator -A 0.5”, see image I.3), had an accuracy of ~43.36% and kappa statistic of 0.0023.

The error on this dataset is 56.64% and the expected error is in the interval (with 95% confidence):

$$48.52\% < error < 64.77\%$$

5.4.3 Rotation Forest

Finally, for the rotation forest, the CVPParameterSelection (image I.4) optimized the minimum and maximum number of groups (G and H) in the ranges 1 to 3 and 3 to 5 respectively. The optimal parameters are $G = 3$ and $H = 5$.

The rotation forest (with parameters “-N -G 3 -H 5 -P 50 -F weka.filters.unsupervised.-attribute.PrincipalComponents -R 1.0 -A 5 -M -1 -S 1 -I 10 -W weka.classifiers.trees.J48 -C 0.25 -M 2”, see image I.5), had an accuracy of ~41.96% and kappa statistic of -0.0153.

The error on this dataset is 58.04% and the expected error is in the interval (with 95% confidence):

$$49.95\% < error < 66.13\%$$

5.4.4 Analysis

When classifying in terms of education the results were not promising at all.

The algorithms had an accuracy between 42% and 46%. The kappa statistics were very small, the best around 0.095 and one even negative. Thus, beside one, the classifiers were not better than classifying randomly.

The confusion matrix shows a lot of confusion between “Bachelor’s degree” and “Master’s degree”. It most probably originated from the unbalanced dataset - 86,9% of the users were in one of the 2 categories.

By calculating the real difference (d) in errors between the algorithms, we reached the following intervals, with the given degree of confidence.

RBFNetwork - Bayesian network

$$-0.143184 < d < 0.0872402(95\%)$$

$$-0.103213 < d < 0.0472687(80\%)$$

RBFNetwork - RotationForest

$$-0.156931 < d < 0.0730145(95\%)$$

$$-0.117042 < d < 0.0331261(80\%)$$

Bayesian network - RotationForest

$$-0.128615 < d < 0.100643(95\%)$$

$$-0.0888459 < d < 0.0608739(80\%)$$

We cannot distinguish the best performing algorithm with a good degree of confidence.

The used classifiers were barely better than chance.

5.5 Conclusion

The classification of the input on age, gender and education was mostly inconclusive.

As said, because of the low amount of input data and unbalanced representation of the population, the results are not to be trusted.

Still, almost all the algorithms were able to predict with accuracy better than by chance, except for classifying education. While classifying gender, the algorithms were slightly better.

This brings hope to future tests, when more data is available to study.

Chapter 6

Conclusion

Through this thesis I analysed the hypothesis of whether it would be possible to relate online users' behaviour with the users' demographics. The purpose was to create an unobtrusive system of tracking and analysis that would allow businesses to better understand their customers and better serve them.

For that, I set out to gather, manipulate and analyse the data of 145 online users who responded to an online questionnaire over a 5-day period.

For analysing the data, several well-known supervised learning algorithms were used for both classification and regression on the demographics of age, gender and education.

The analysis of whether said relation exists was inconclusive.

The creation of the proposed system depends on the success of predicting demographics, so I will discuss how the analysis can be improved and better tested in the future.

6.1 Future steps

As mentioned throughout the analysis, the biggest problem is the low number and the imbalance present in the instances gathered to train the machine learning model. The first step would therefore be to obtain a larger, more diverse dataset that would enable better predictions.

Another possible improvement would be to try other machine learning algorithms that could categorise/model the data more effectively.

Finally, the analysis could be more complete by considering more input variables or more demographics.

In addition to those variables used in this analysis, there is a multitude of other potential browsing behaviour variables that could be used, specially oriented to mobile or tablet usage.

In this analysis the data gathering did not capture much of the users' possible behaviour, for example related to typing or interacting with richer media. Access to a more complete website, higher traffic and users for longer sessions would have been helpful in this regard.

Furthermore, while I only studied how those behaviours related to three demographics, maybe others can be better related, such as occupation, social status, country/culture, and so on.

Some of these improvements were limitations identified ex-ante; while others were not applied because this would massively increase the scope of this exercise.

Nevertheless, there are improvements I plan to make in the near future, in order to obtain more robust and reliable estimates that will allow me to further develop this theory.

Appendix A

Repository

The screenshot shows a GitHub repository page for a private repository named 'lib' owned by the user 'fil090302'. The repository has 67 commits, 1 branch, 0 releases, and 1 contributor. The current branch is 'master'. The repository is private, as indicated by the lock icon and the 'PRIVATE' label. The main content area displays a list of files and their commit history:

File	Description	Time
api	Added other interests property	2 months ago
modules	first commit	8 months ago
public/js	Small fix	2 months ago
schema	Added other interests property	2 months ago
setup	Minor fix to allow only wanted origins	2 months ago
views	Added input to index	2 months ago
.gitignore	First version of lib, init event completed with all possibilities che...	7 months ago
Procfile	first commit	8 months ago
app.js	Attempt to upgrade to express 4	2 months ago
models.js	Added functionality to save questionnaire information. (To remove)	2 months ago
package.json	Minor fix	2 months ago
routes.js	Added functionality to save questionnaire information. (To remove)	2 months ago

On the right side, there are navigation links for Code, Issues (0), Pull requests (0), Wiki, Pulse, Graphs, and Settings. At the bottom right, there are options to clone the repository using SSH (git@github.com:file) or HTTPS, and buttons for 'Clone in Desktop' and 'Download ZIP'.

Figure A.1: The private repository with the developed system (ask for permission: filipmanuel.lp@gmail.com).

Appendix B

Events

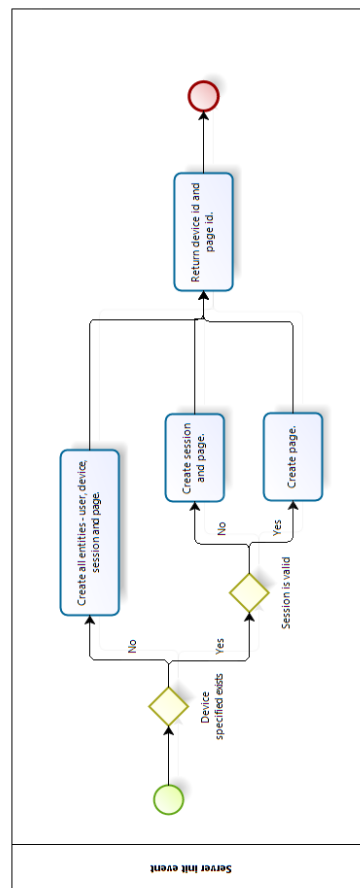


Figure B.1: The logic path taken by the server in incoming Init events.

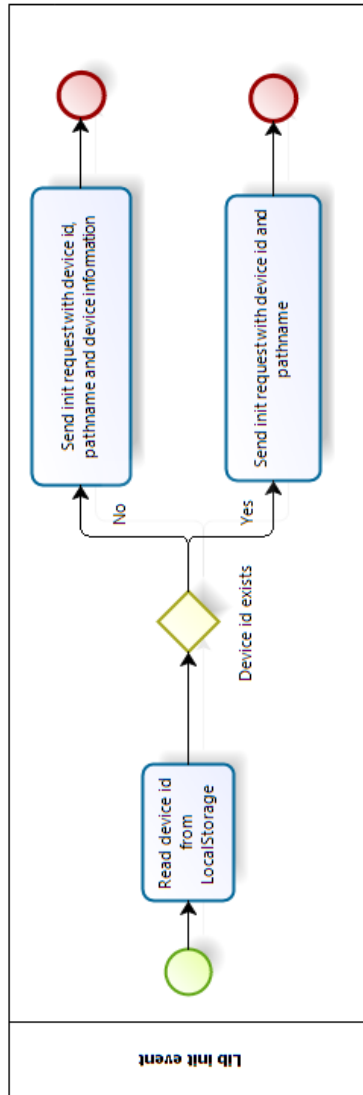


Figure B.2: The logic path taken by the library in sending Init events.

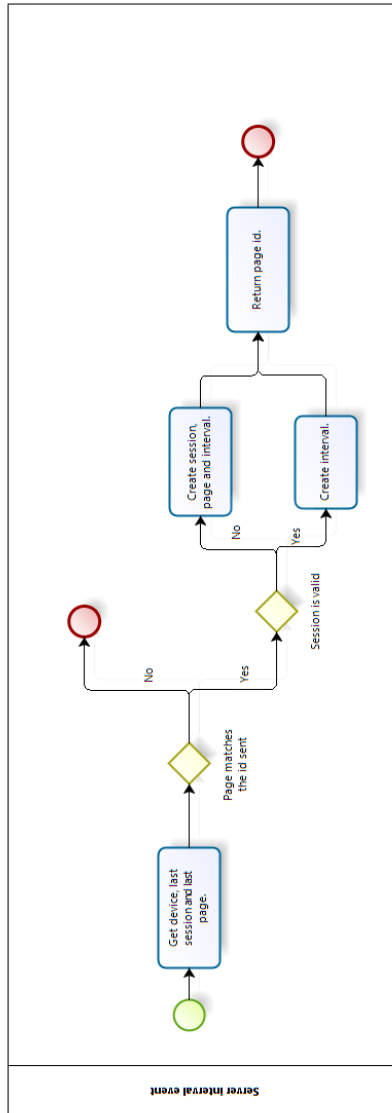


Figure B.3: The logic path taken by the server in incoming Interval events.

Appendix C

Questionnaire screenshots

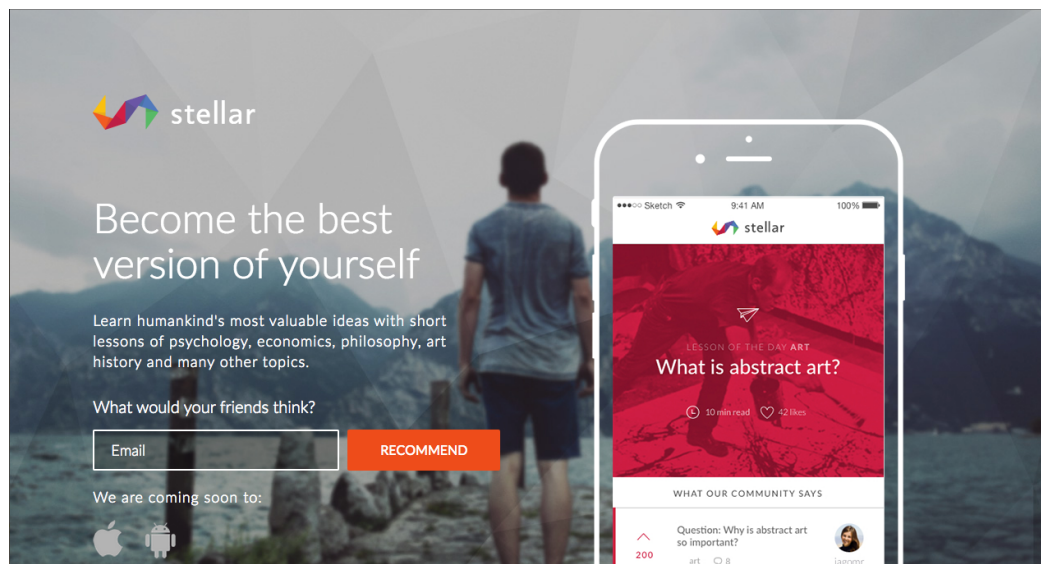


Figure C.1: The website used for the questionnaire.

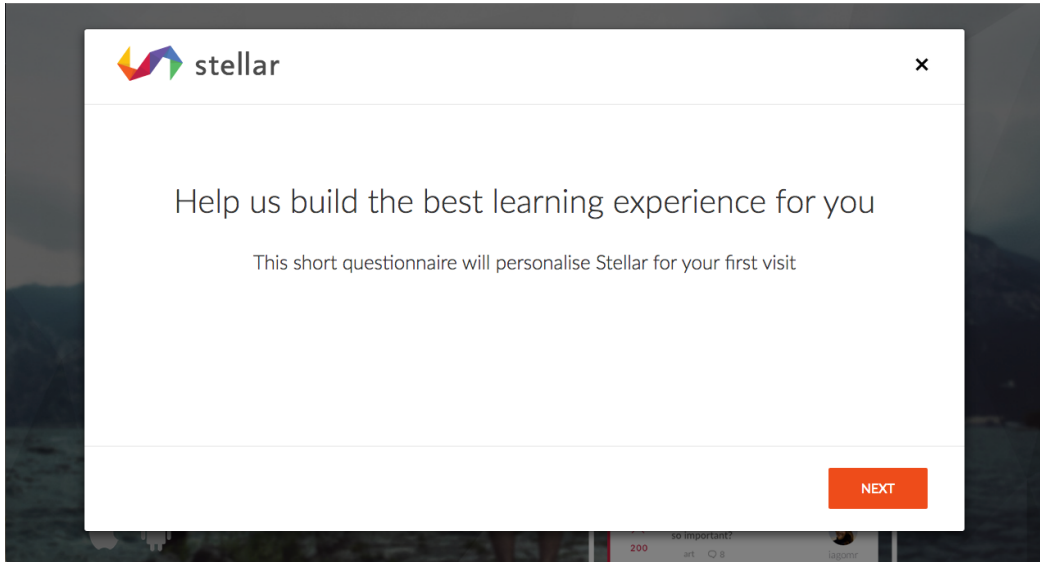


Figure C.2: The intro to the questionnaire.

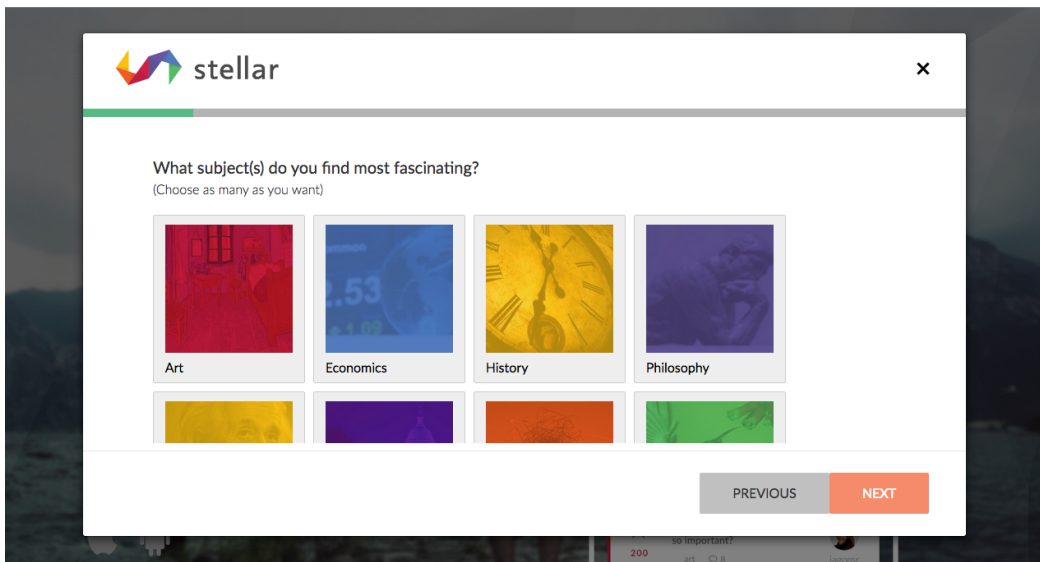


Figure C.3: The first step of the questionnaire.

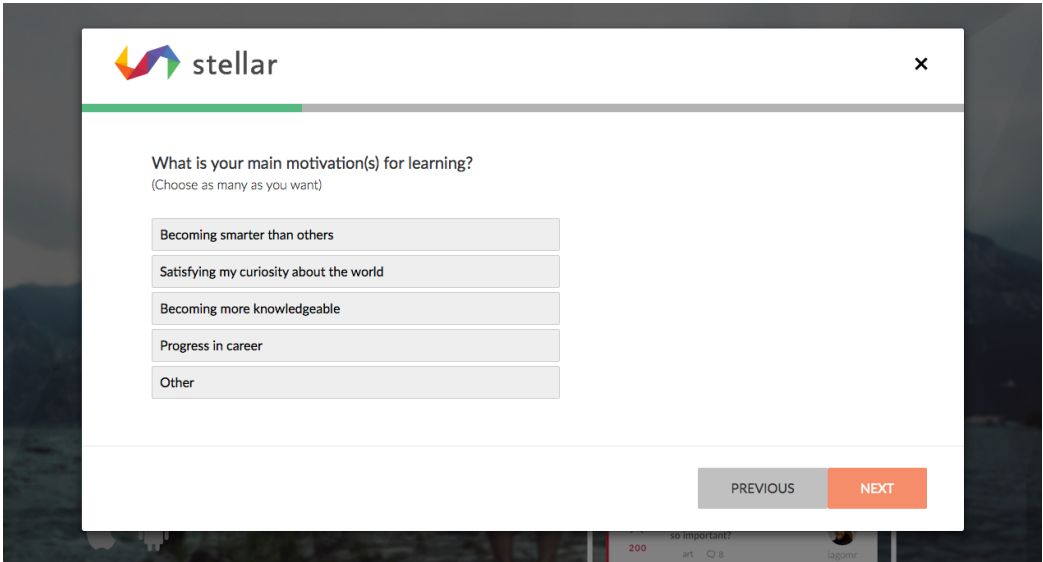


Figure C.4: The second step of the questionnaire.

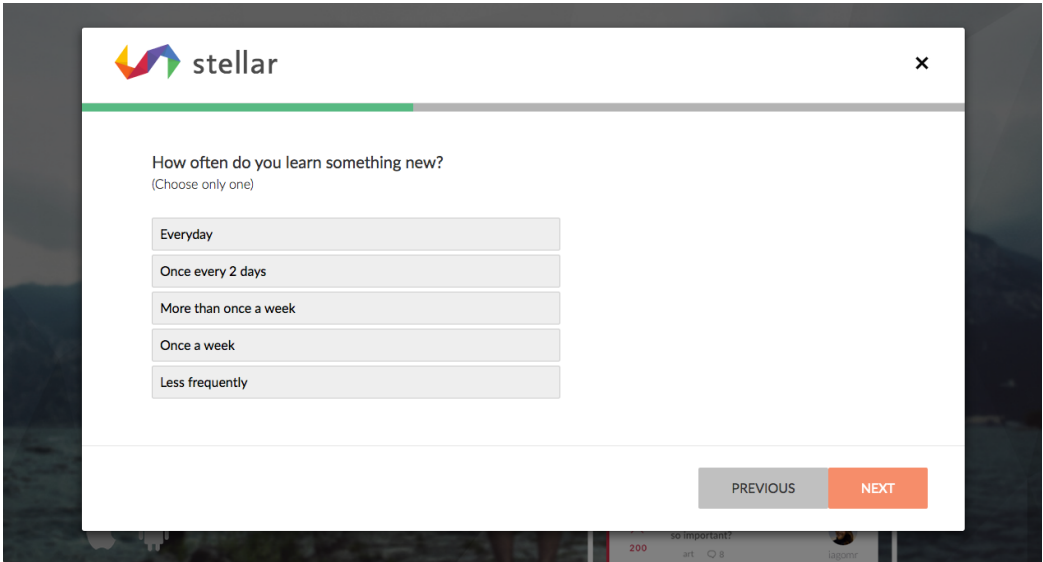


Figure C.5: The third step of the questionnaire.

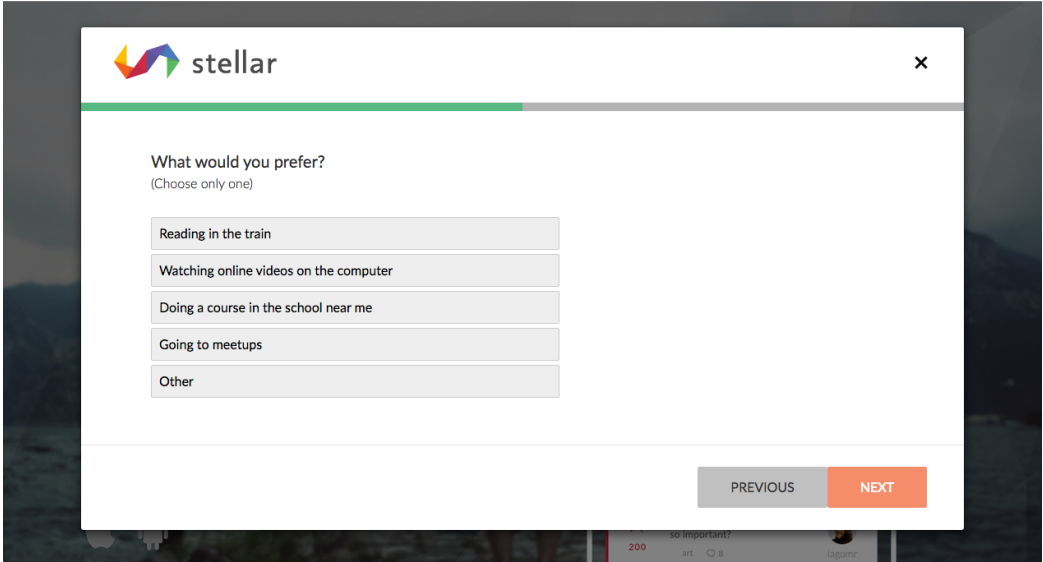


Figure C.6: The fourth step of the questionnaire.

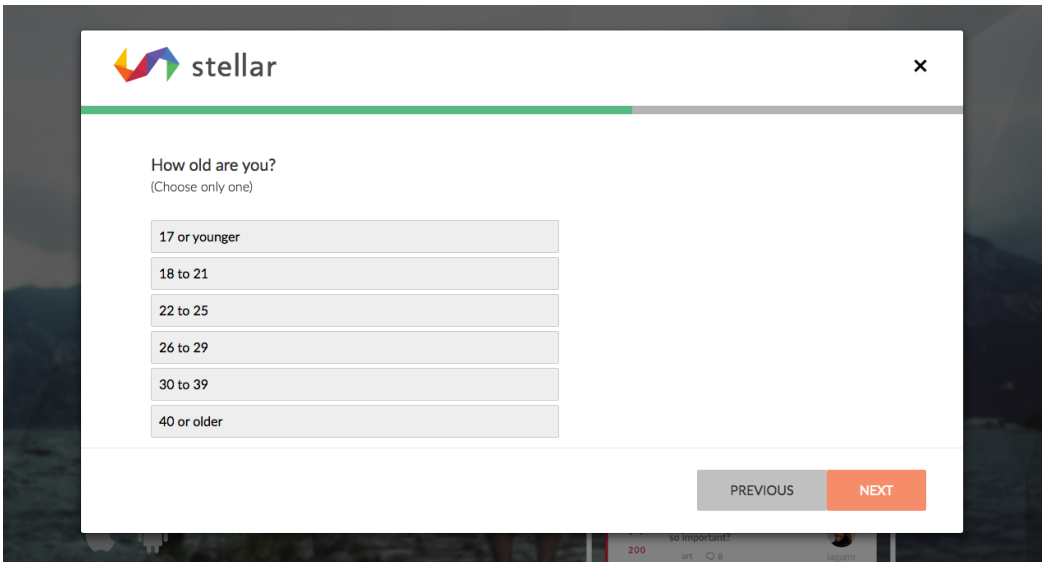


Figure C.7: The fifth step of the questionnaire.

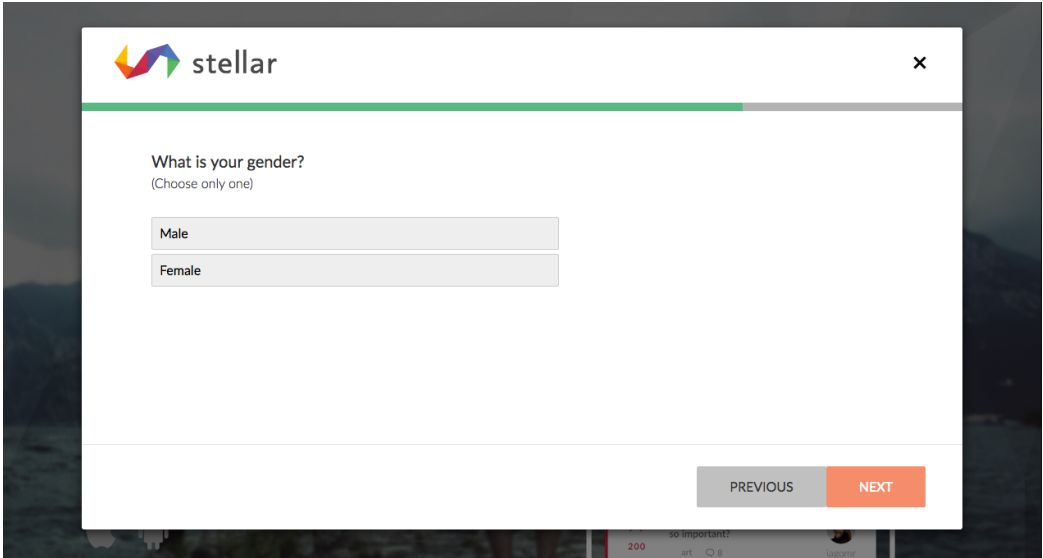


Figure C.8: The sixth step of the questionnaire.

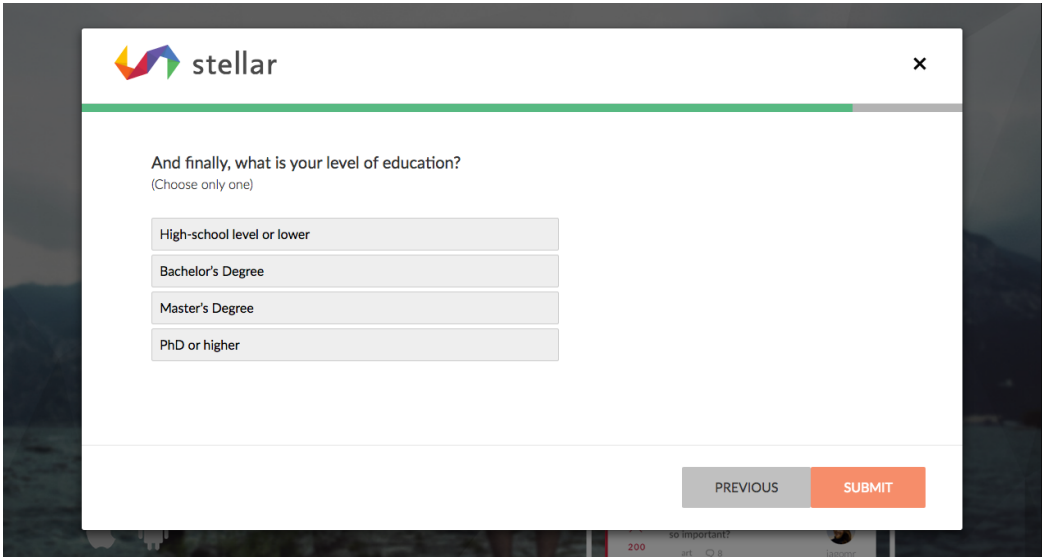


Figure C.9: The seventh step of the questionnaire.

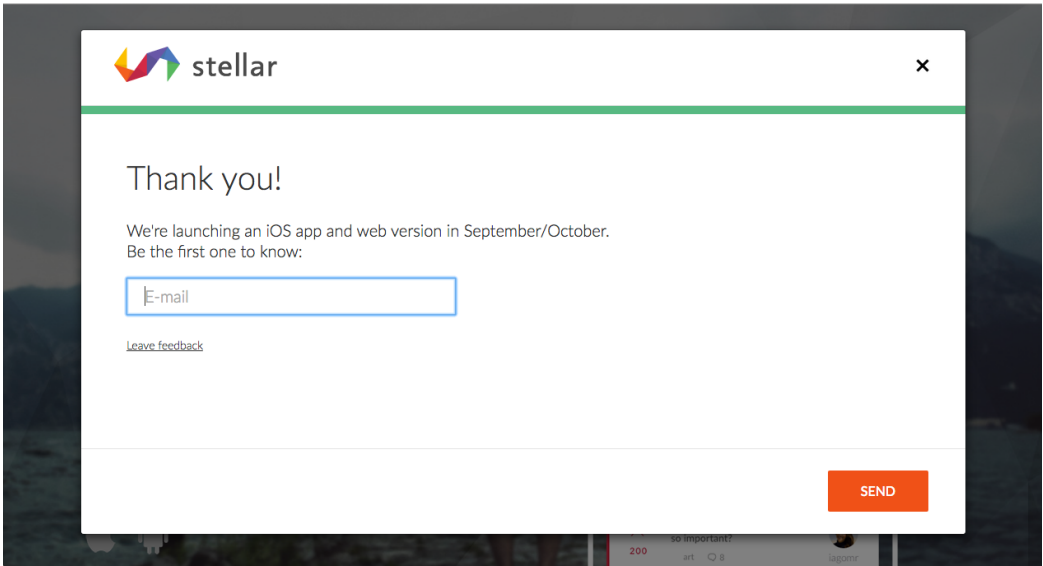


Figure C.10: The success page of the questionnaire.

Appendix D

Facebook ads

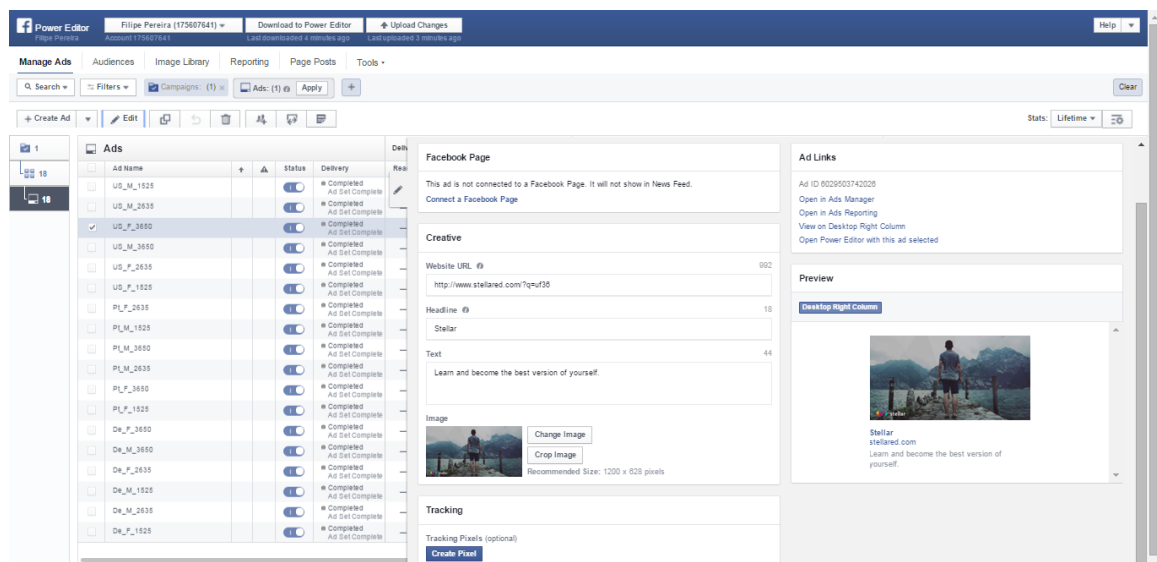
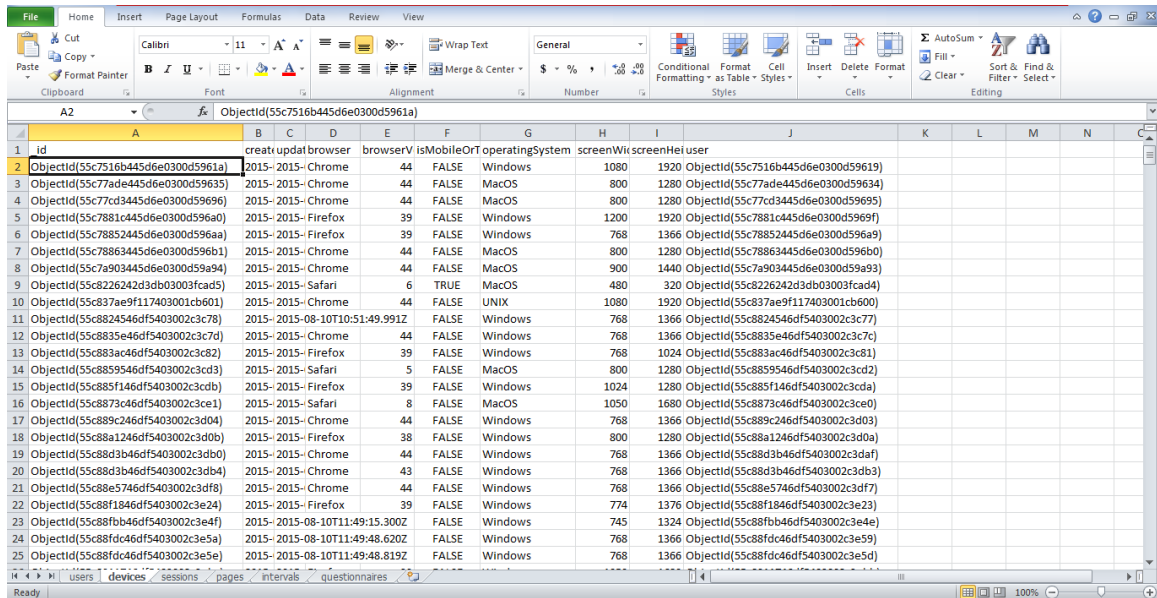


Figure D.1: Overview of the facebook ads used to gather users.

Appendix E

Data gathered



id	creat	updat	browser	browserV	isMobileOrT	operatingSystem	screenW	screenHei	user
Objectid(55c7516b445d6e0300d5961a)	2015-	2015-	Chrome	44	FALSE	Windows	1080	1920	Objectid(55c7516b445d6e0300d59619)
Objectid(55c77ade445d6e0300d59635)	2015-	2015-	Chrome	44	FALSE	MacOS	800	1280	Objectid(55c77ade445d6e0300d59634)
Objectid(55c77cd3445d6e0300d59696)	2015-	2015-	Chrome	44	FALSE	MacOS	800	1280	Objectid(55c77cd3445d6e0300d59695)
Objectid(55c7881c445d6e0300d596a0)	2015-	2015-	Firefox	39	FALSE	Windows	1200	1920	Objectid(55c7881c445d6e0300d5969f)
Objectid(55c78852445d6e0300d596aa)	2015-	2015-	Firefox	39	FALSE	Windows	768	1366	Objectid(55c78852445d6e0300d596a9)
Objectid(55c78863445d6e0300d596b1)	2015-	2015-	Chrome	44	FALSE	MacOS	800	1280	Objectid(55c78863445d6e0300d596b0)
Objectid(55c7a903445d6e0300d59a94)	2015-	2015-	Chrome	44	FALSE	MacOS	900	1440	Objectid(55c7a903445d6e0300d59a93)
Objectid(55c8226242d3db03003fca05)	2015-	2015-	Safari	6	TRUE	MacOS	480	320	Objectid(55c8226242d3db03003fca04)
Objectid(55c837ae9f117403001cb601)	2015-	2015-	Chrome	44	FALSE	UNIX	1080	1920	Objectid(55c837ae9f117403001cb600)
Objectid(55c8824546df5403002c3c78)	2015-	2015-08-10T10:51:49.991Z	Chrome	44	FALSE	Windows	768	1366	Objectid(55c8824546df5403002c3c77)
Objectid(55c8835e46df5403002c3c7d)	2015-	2015-	Chrome	44	FALSE	Windows	768	1366	Objectid(55c8835e46df5403002c3c7c)
Objectid(55c883ac46df5403002c3c82)	2015-	2015-	Firefox	39	FALSE	Windows	768	1024	Objectid(55c883ac46df5403002c3c81)
Objectid(55c8859546df5403002c3cd3)	2015-	2015-	Safari	5	FALSE	MacOS	800	1280	Objectid(55c8859546df5403002c3cd2)
Objectid(55c885f146df5403002c3cdb)	2015-	2015-	Firefox	39	FALSE	Windows	1024	1280	Objectid(55c885f146df5403002c3cda)
Objectid(55c8873c46df5403002c3ce1)	2015-	2015-	Safari	8	FALSE	MacOS	1050	1680	Objectid(55c8873c46df5403002c3ce0)
Objectid(55c889c246df5403002c3d04)	2015-	2015-	Chrome	44	FALSE	Windows	768	1366	Objectid(55c889c246df5403002c3d03)
Objectid(55c88a1246df5403002c3d0b)	2015-	2015-	Firefox	38	FALSE	Windows	800	1280	Objectid(55c88a1246df5403002c3d0a)
Objectid(55c88d3b46df5403002c3db0)	2015-	2015-	Chrome	44	FALSE	Windows	768	1366	Objectid(55c88d3b46df5403002c3daf)
Objectid(55c88d3b46df5403002c3db4)	2015-	2015-	Chrome	43	FALSE	Windows	768	1366	Objectid(55c88d3b46df5403002c3db3)
Objectid(55c88e5746df5403002c3df8)	2015-	2015-	Chrome	44	FALSE	Windows	768	1366	Objectid(55c88e5746df5403002c3df7)
Objectid(55c88f1846df5403002c3e24)	2015-	2015-	Firefox	39	FALSE	Windows	774	1376	Objectid(55c88f1846df5403002c3e23)
Objectid(55c88fbb46df5403002c3e4f)	2015-	2015-08-10T11:49:15.300Z	Chrome	44	FALSE	Windows	745	1324	Objectid(55c88fbb46df5403002c3e4e)
Objectid(55c88fdd46df5403002c3e5a)	2015-	2015-08-10T11:49:48.620Z	Chrome	44	FALSE	Windows	768	1366	Objectid(55c88fdd46df5403002c3e59)
Objectid(55c88fd46df5403002c3e5e)	2015-	2015-08-10T11:49:48.819Z	Chrome	44	FALSE	Windows	768	1366	Objectid(55c88fd46df5403002c3e5d)

Figure E.1: A screenshot of some of the data gathered.

Appendix F

Scripts for data manipulation

```
1  /**
2  * Scrolling data manipulation.
3  */
4
5  function getRange(inputs) {
6
7      var max = 0;
8
9      inputs.forEach(function(input) {
10         if(parseInt(input.y) > max) {
11             max = parseInt(input.y);
12         }
13     });
14
15     return max;
16 }
17
18 function getDistanceScrolled(inputs) {
19
20     var previousY = parseInt(inputs[0].y);
21     var distanceSum = 0;
22
23     // Remove first element.
24     inputs.shift();
25
26     inputs.forEach(function(input) {
27
28         distanceSum += Math.abs(parseInt(input.y) - previousY);
29
30         previousY = parseInt(input.y);
31     });
32
33     return distanceSum;
34 }
35
36 // Speed from timestamps.
37 function getAverageSpeed(inputs) {
38
39     var previousTimeStamp = parseInt(inputs[0].timestamp);
40     var previousY = parseInt(inputs[0].y);
41     var speedSum = 0;
42
43     // Remove first element.
44     inputs.shift();
```

Figure F.1: A screenshot of some of the JS functions I used to manipulate the gathered data.

Appendix G

Classification on age

G.1 CVPParameterSelection for the RBFNetwork model

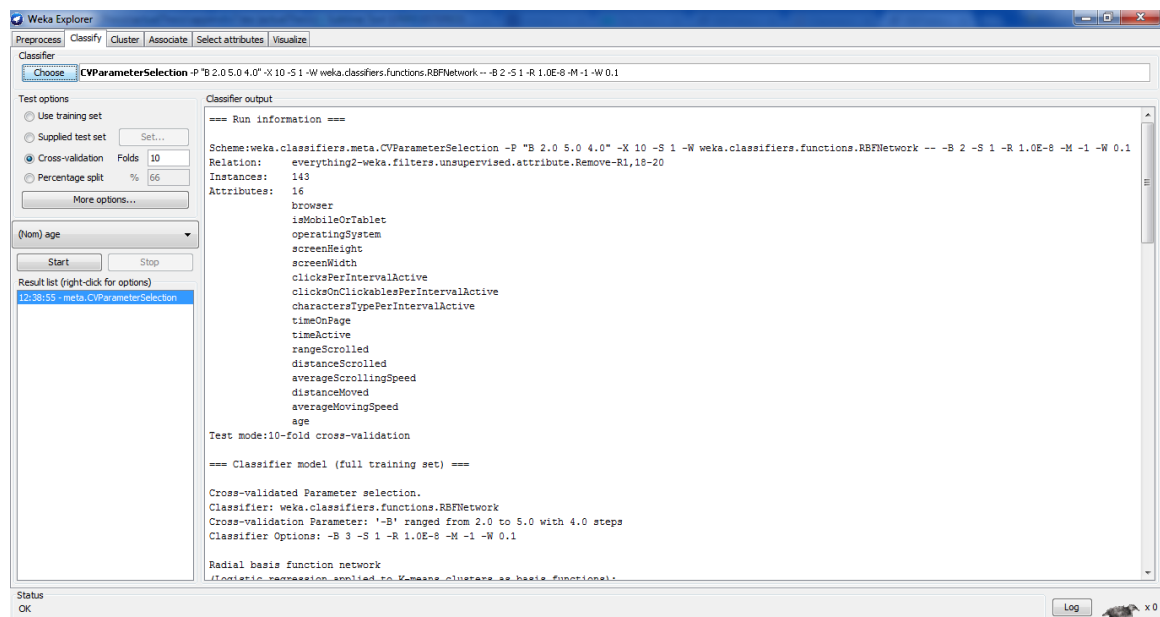


Figure G.1: A screenshot of the results of using the CVPParameterSelection for parameter optimization on the RBFNetwork.

G.2 RBFNetwork model

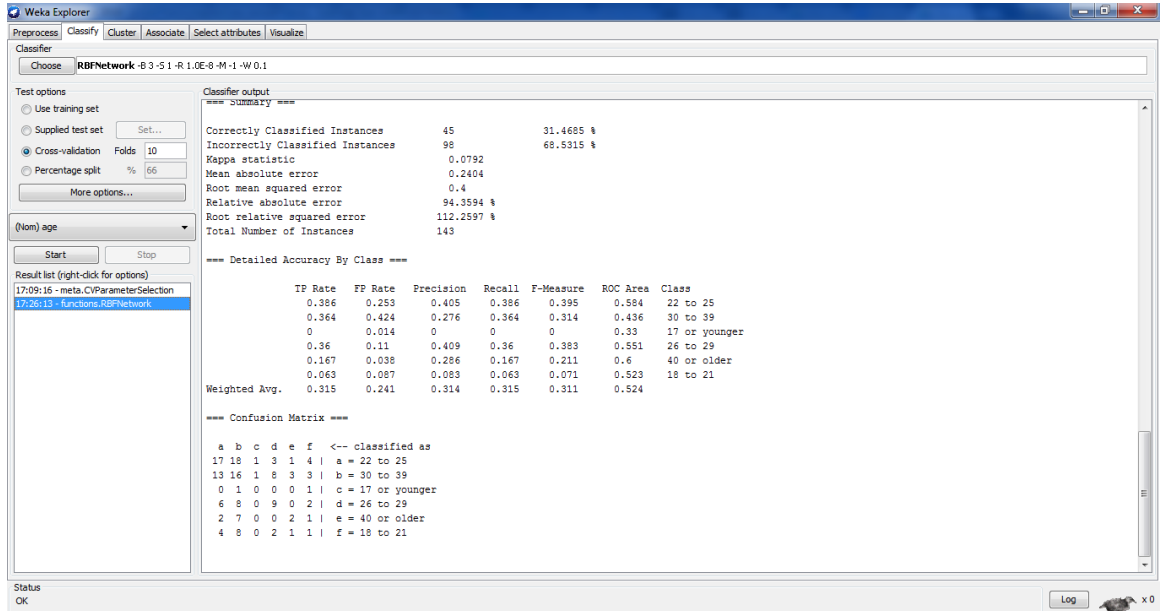


Figure G.2: A screenshot of the results of using the RBFNetwork method to classify on age.

G.3 Bayesian network model

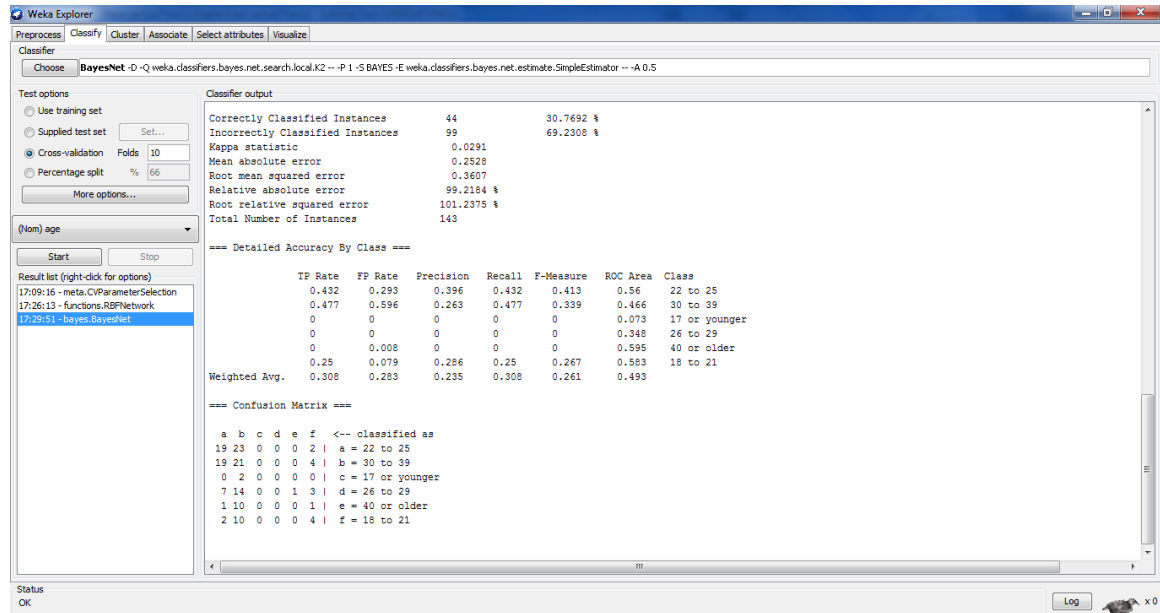


Figure G.3: A screenshot of the results of using the Bayesian network method to classify on age.

G.4 CVParameterSelection for the Rotation Forest model

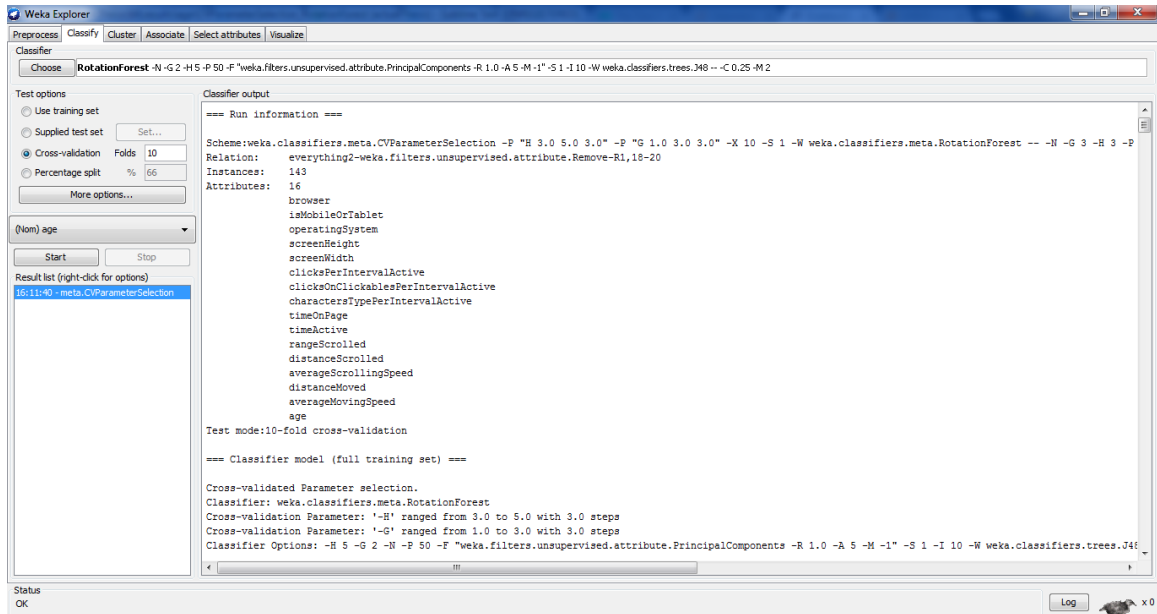


Figure G.4: A screenshot of the results of using the CVParameterSelection for parameter optimization on the Rotation forest method.

G.5 Rotation Forest model

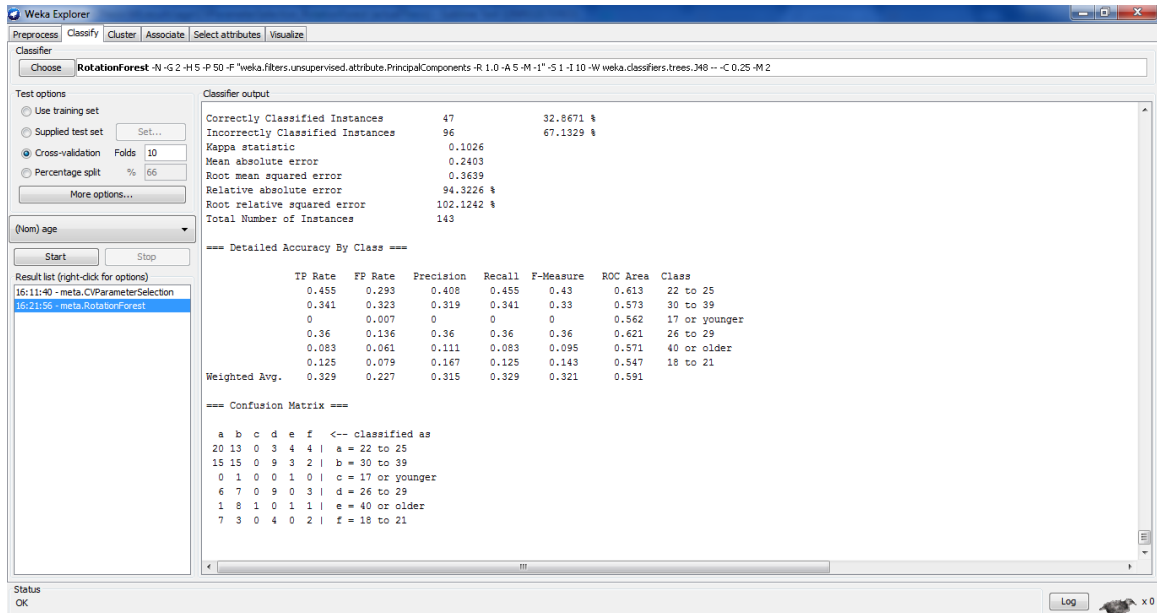


Figure G.5: A screenshot of the results of using the Rotation forest method to classify on age.

Appendix H

Classification on gender

H.1 CVPParameterSelection for the RBFNetwork model

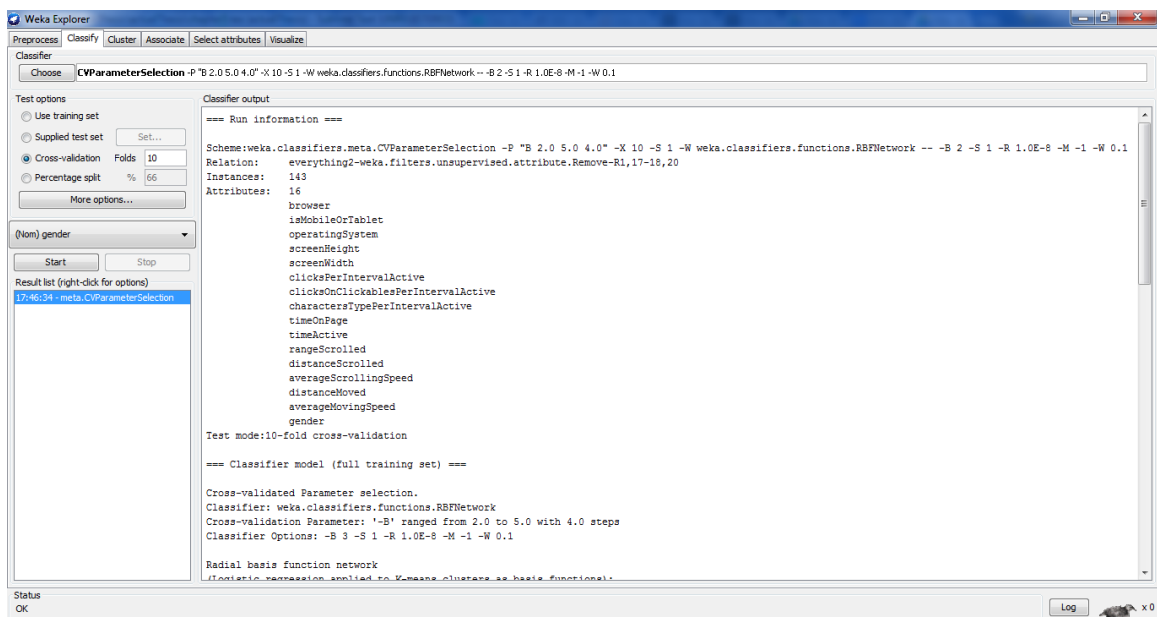


Figure H.1: A screenshot of the results of using the CVPParameterSelection for parameter optimization on the RBFNetwork method.

H.2 RBFNetwork model

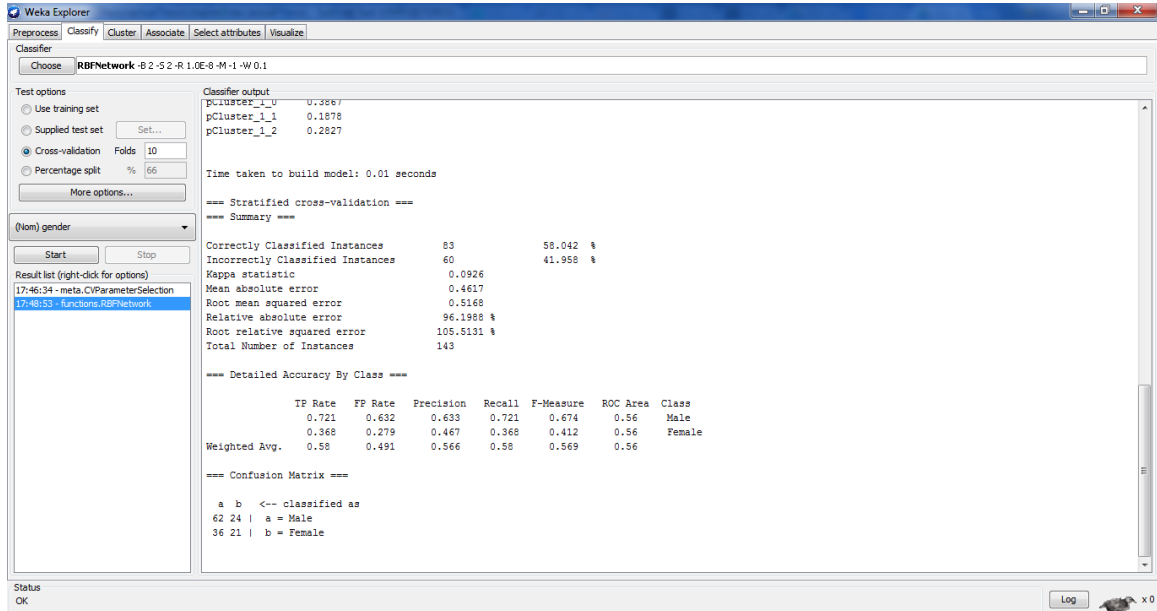


Figure H.2: A screenshot of the results of using the RBFNetwork method to classify on gender.

H.3 Bayesian network model

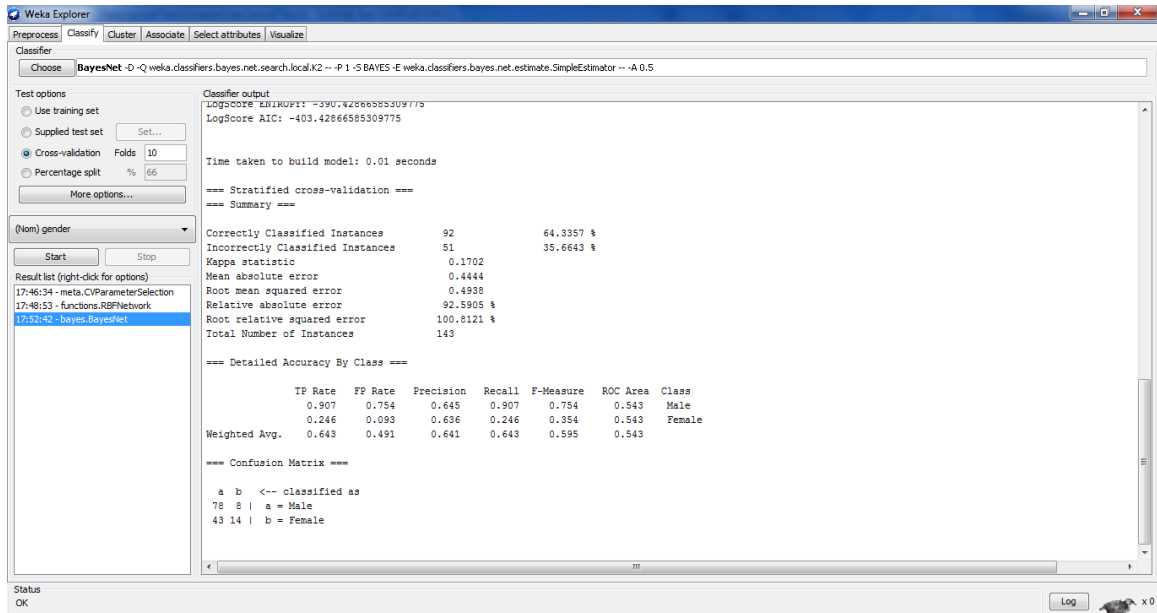


Figure H.3: A screenshot of the results of using the Bayesian network method to classify on gender.

H.4 CVPParameterSelection for the Rotation Forest model

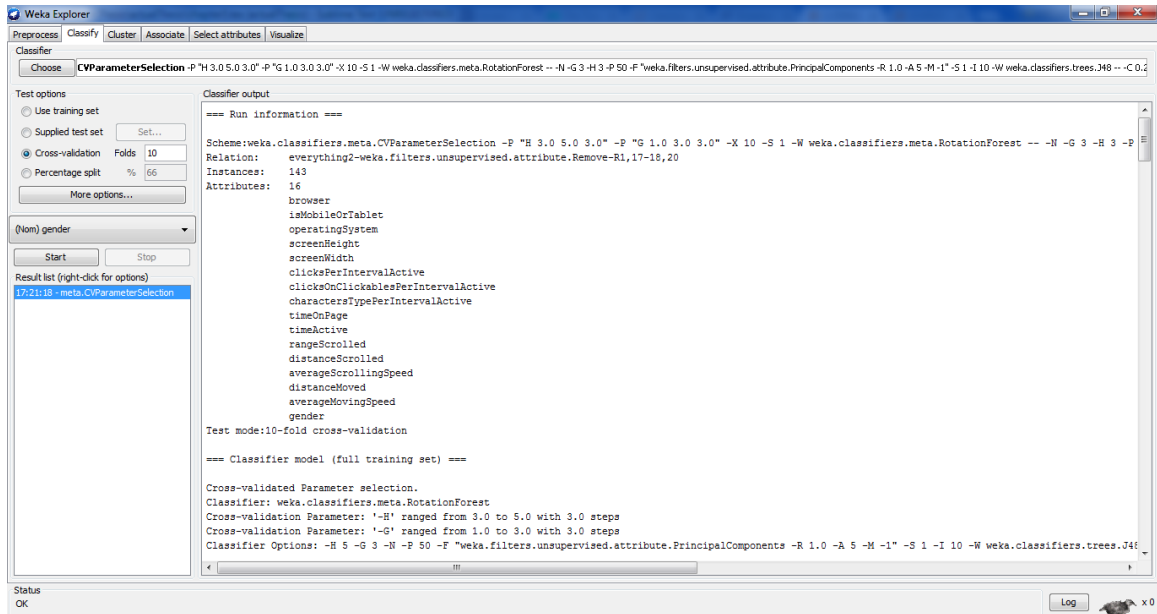


Figure H.4: A screenshot of the results of using the CVPParameterSelection for parameter optimization on the Rotation forest method.

H.5 Rotation Forest model

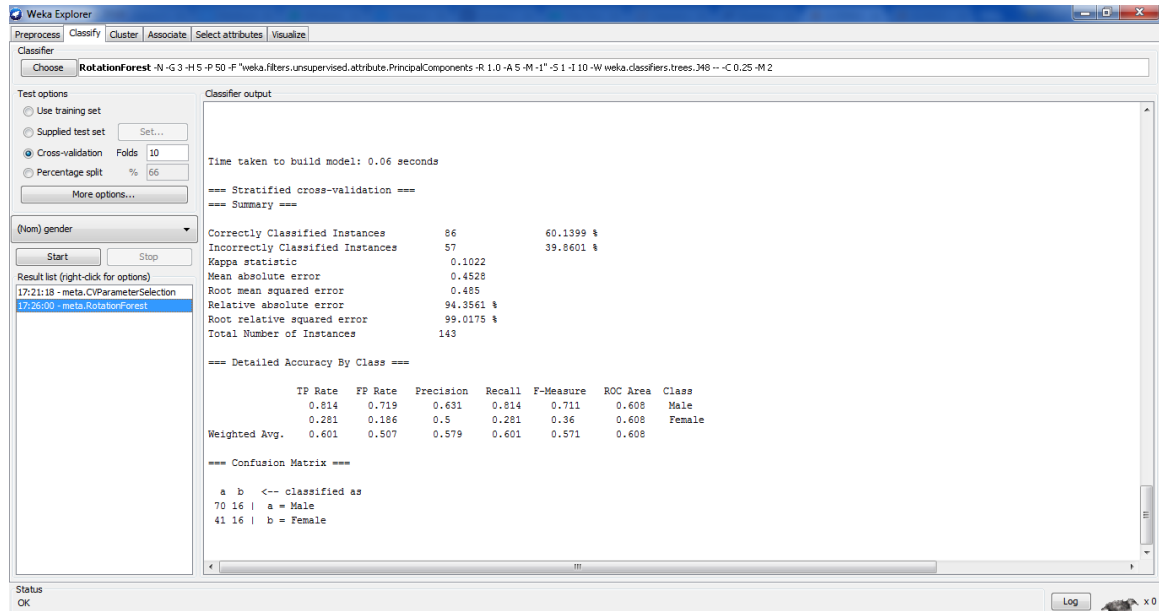


Figure H.5: A screenshot of the results of using the Rotation forest method to classify on gender.

Appendix I

Classification on education

I.1 CVPParameterSelection for the RBFNetwork model

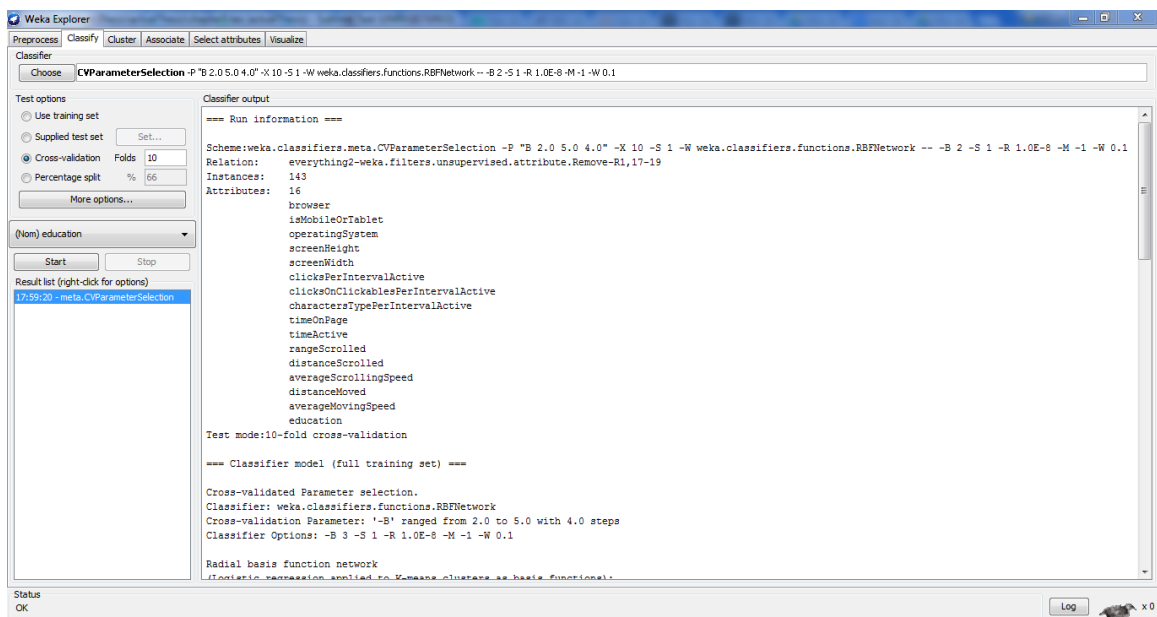


Figure I.1: A screenshot of the results of using the CVPParameterSelection for parameter optimization on the RBFNetwork method.

I.2 RBFNetwork model

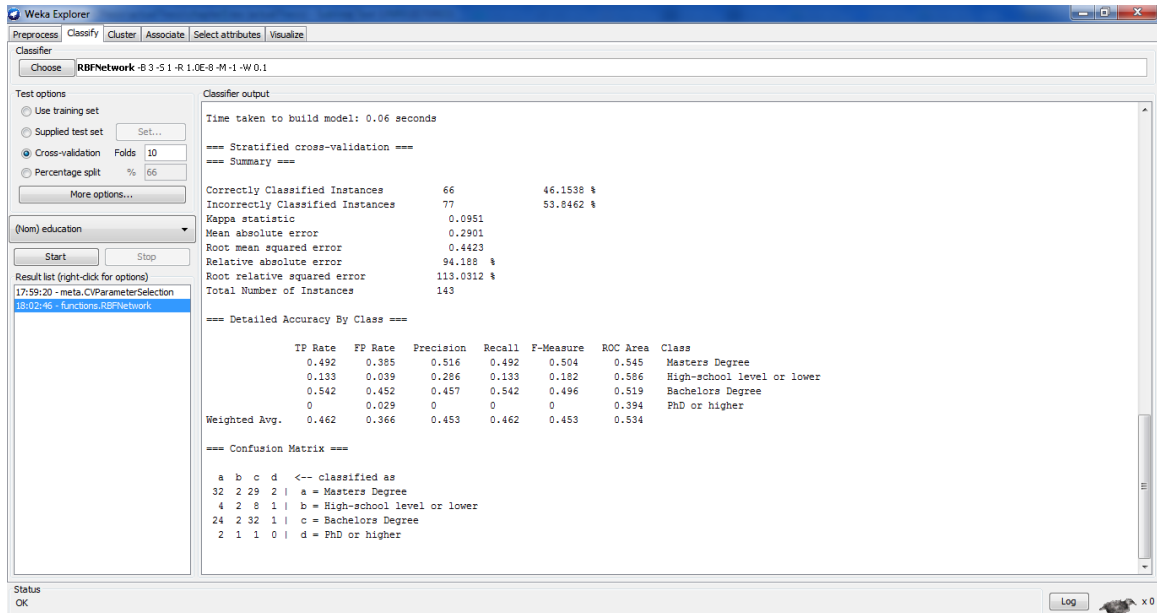


Figure I.2: A screenshot of the results of using the RBFNetwork method to classify on education.

I.3 Bayesian network model

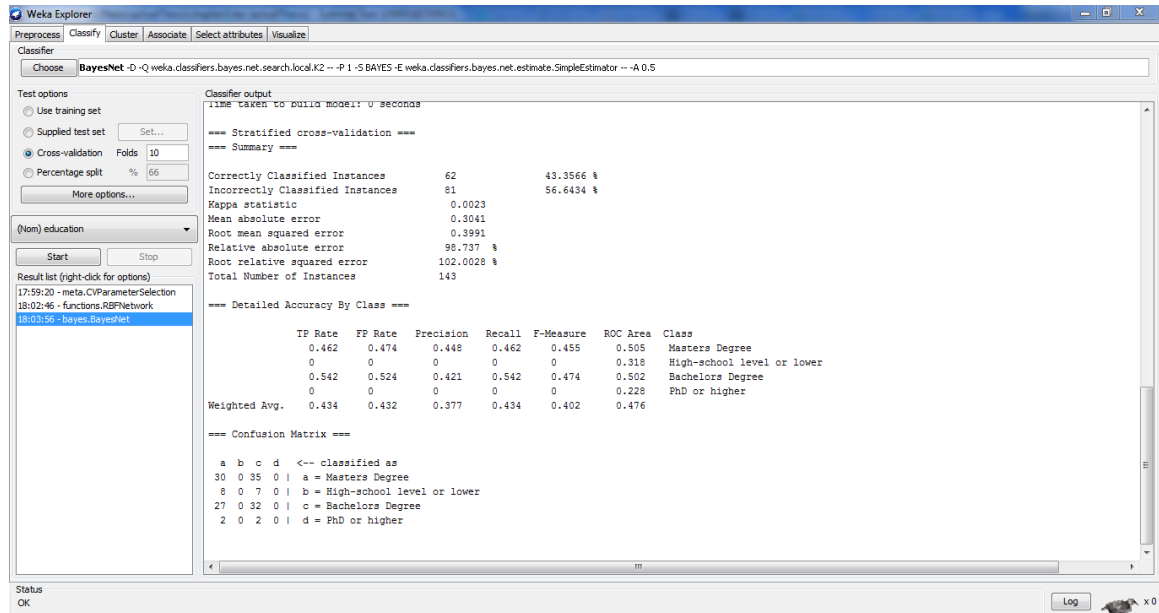


Figure I.3: A screenshot of the results of using the Bayesian network method to classify on education.

I.4 CVPParameterSelection for the Rotation Forest model

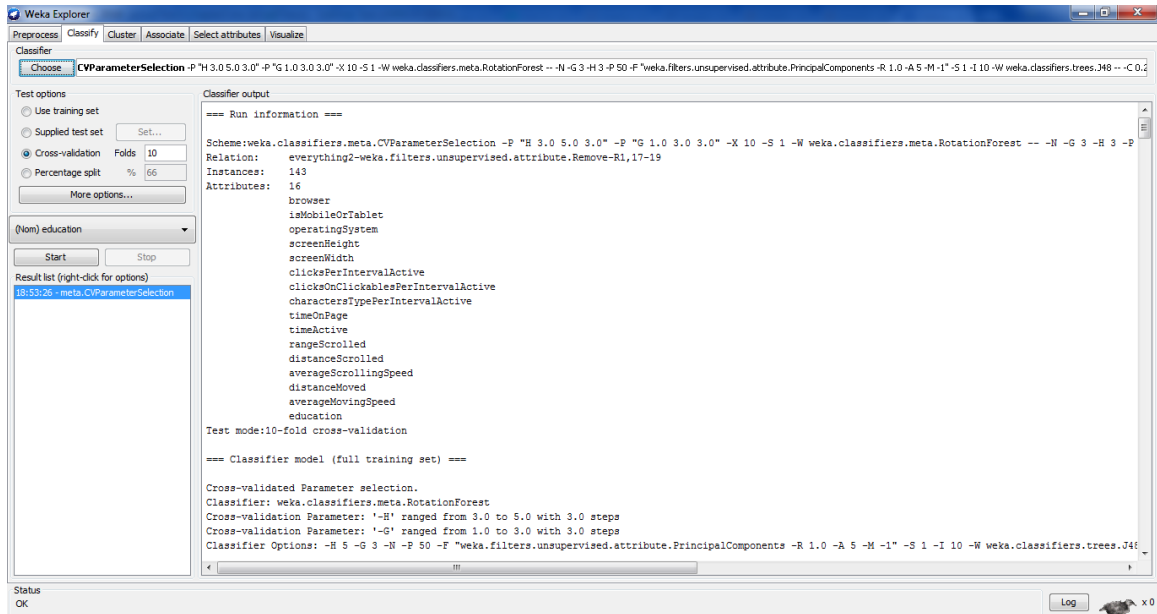


Figure I.4: A screenshot of the results of using the CVPParameterSelection for parameter optimization on the Rotation forest method.

I.5 Rotation Forest model

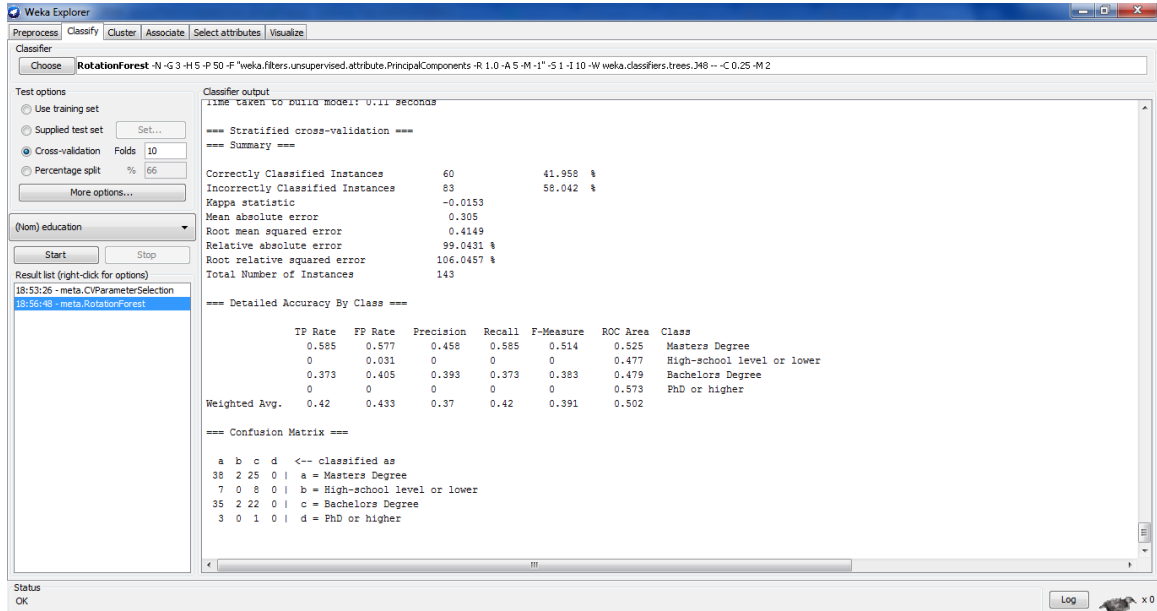


Figure I.5: A screenshot of the results of using the Rotation forest method to classify on education.

Appendix J

Regression on age

J.1 Gaussian processes

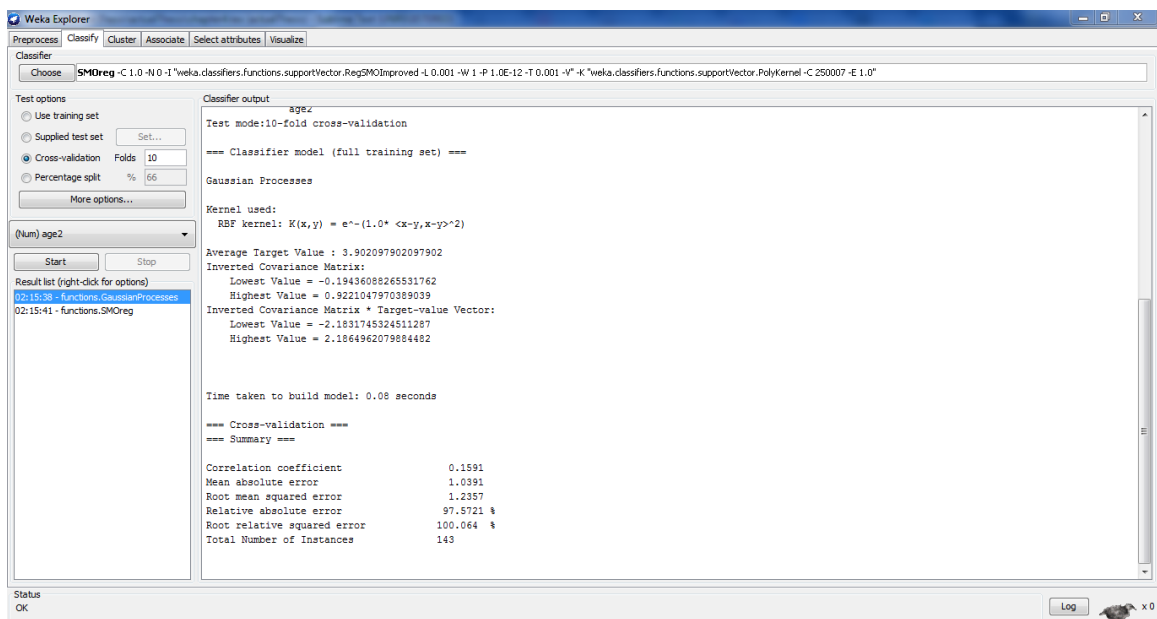


Figure J.1: A screenshot of the results of using Gaussian processes as a regression algorithm on age.

J.2 SMOreg

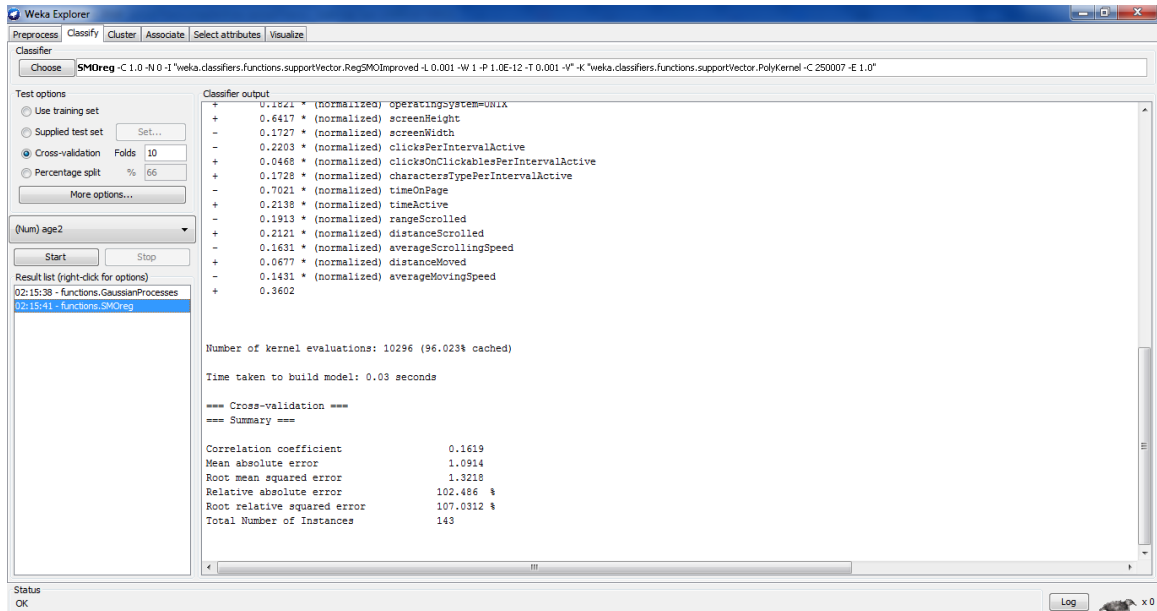


Figure J.2: A screenshot of the results of using SMOreg as a regression algorithm on age.

Bibliography

- [1] “Convolutional neural networks for visual recognition.” <http://cs231n.github.io/convolutional-networks/>. [Online; accessed September-2015].
- [2] “Bayesian network image.” <http://www.niedermayer.ca/book/export/html/25>. [Online; accessed September-2015].
- [3] “Decision tree image.” <http://www.doc.ic.ac.uk/~sgc/teaching/pre2012/v231/lecture11.html>. [Online; accessed September-2015].
- [4] “Confusion matrix image.” <http://i.stack.imgur.com/ysM0Z.png>. [Online; accessed September-2015].
- [5] “Normal distribution.” https://en.wikipedia.org/wiki/Normal_distribution. [Online; accessed September-2015].
- [6] “Explanation of the gaussian processes.” <http://qr.ae/RPqxd>. [Online; accessed September-2015].
- [7] “Support vector machines (svm) introductory overview.” <http://www.statsoft.com/Textbook/Support-Vector-Machines>. [Online; accessed September-2015].
- [8] “Correlation.” <http://www.mathsisfun.com/data/correlation.html>. [Online; accessed September-2015].
- [9] “Internet Live Stats - internet users.” <http://www.internetlivestats.com/internet-users/>. [Online; accessed September-2015].
- [10] “Facebook ads.” <https://www.facebook.com/business/products/ads>. [Online; accessed September-2015].
- [11] “Google adwords.” <https://www.google.com/adwords/>. [Online; accessed September-2015].

- [12] “Google, Yahoo, Facebook team up in fight against bad ad bots.” [http : / / www . zdnet . com / article / google-yahoo-facebook-team-up-in-fight-against-bad-ad-bots/](http://www.zdnet.com/article/google-yahoo-facebook-team-up-in-fight-against-bad-ad-bots/). [Online; accessed September-2015].
- [13] “Intercom.” <https://www.intercom.io/>. [Online; accessed September-2015].
- [14] “Google analytics.” <http://www.google.com/analytics/>. [Online; accessed September-2015].
- [15] “Mixpanel.” <https://mixpanel.com/>. [Online; accessed September-2015].
- [16] J. I. Hong, J. Heer, S. Waterson, and J. A. Landay, “Webquilt: A proxy-based approach to remote web usability testing,”
- [17] R. Atterer, M. Wnuk, and A. Schmidt, “Knowing the user’s every move: User activity tracking for website usability evaluation and implicit interaction,” in *Proceedings of the 15th International Conference on World Wide Web*.
- [18] J. Goecks and J. Shavlik, “Learning users’ interests by unobtrusively observing their normal behavior,”
- [19] “Userlytics.” <http://www.userlytics.com/sitepublic/>. [Online; accessed September-2015].
- [20] “Userzoom.” <http://www.userzoom.co.uk/>. [Online; accessed September-2015].
- [21] “Hotjar.” <https://www.hotjar.com/>. [Online; accessed September-2015].
- [22] J. Hu, H.-J. Zeng, H. Li, C. Niu, and Z. Chen, “Demographic prediction based on user’s browsing behavior,” in *WWW ’07: Proceedings of the 16th international conference on World Wide Web*, (New York, NY, USA), pp. 151–160, ACM, 2007.
- [23] “About demographics and interests.” <https://support.google.com/analytics/answer/2799357>. [Online; accessed September-2015].
- [24] “Demographics and interests data collection and thresholds.” <https://support.google.com/analytics/answer/2954071>. [Online; accessed September-2015].

- [25] “<http://www.humix.be/en/blog/how-accurate-are-google-analytics-demographics-reports>.” [http : / / www . humix . be / en / blog / how-accurate-are-google-analytics-demographics-reports](http://www.humix.be/en/blog/how-accurate-are-google-analytics-demographics-reports). [Online; accessed September-2015].
- [26] R. Jones, R. Kumar, B. Pang, and A. Tomkins, ““i know what you did last summer”: query logs and user privacy,” in *CIKM '07: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pp. 909–914, ACM, 2007.
- [27] S. Goel, J. M. Hofman, and M. I. Siner, “Who does what on the web: A large-scale study of browsing behavior,” in *ICWSM* (J. G. Breslin, N. B. Ellison, J. G. Shanahan, and Z. Tufekci, eds.), The AAAI Press, 2012.
- [28] K. De Bock, D. Van den Poel, and S. Manigart, “Predicting web site audience demographics for web advertising targeting using multi-web site clickstream data,” working papers of faculty of economics and business administration, ghent university, belgium, Ghent University, Faculty of Economics and Business Administration, 2009.
- [29] E. Hargittai, “Whose space? differences among users and non-users of social network sites,”
- [30] D. Boyd and B. University of California, *Taken Out of Context: American Teen Sociality in Networked Publics*. University of California, Berkeley, 2008.
- [31] M. Abramson and D. W. Aha, “User authentication from web browsing behavior.” <https://www.aaai.org/ocs/index.php/FLAIRS/FLAIRS13/paper/viewFile/5865/6081>.
- [32] E. Shi, Y. Niu, M. Jakobsson, and R. Chow, “Implicit authentication through learning user behavior,”
- [33] “Mobile marketing statistics 2015.” [http : / / www . smartinsights . com / mobile-marketing / mobile-marketing-analytics / mobile-marketing-statistics/](http://www.smartinsights.com/mobile-marketing/mobile-marketing-analytics/mobile-marketing-statistics/). [Online; accessed September-2015].
- [34] “Crashlytics.” <http://try.crashlytics.com/>. [Online; accessed September-2015].
- [35] “Flurry.” <http://www.flurry.com/solutions/analytics>. [Online; accessed September-2015].

- [36] G. I. Webb, M. J. Pazzani, and D. Billsus, “Machine learning for user modeling,” *User Modeling and User-Adapted Interaction*.
- [37] “Concept drift.” https://en.wikipedia.org/wiki/Concept_drift. [Online; accessed Sempteber-2015].
- [38] “Which way is the concept of vertical scrolling headed in 2015?.” <http://www.awwwards.com/which-way-is-the-concept-of-vertical-scrolling-headed-in-2015.html>. [Online; accessed Sempteber-2015].
- [39] “Marketer advising coke, microsoft questions ethics of widespread web tracking.” <http://www.forbes.com/sites/adamtanner/2013/08/26/marketer-advising-coke-microsoft-questions-ethics-of-widespread-web-tracking/>. [Online; accessed Sempteber-2015].
- [40] “Data protection in the eu.” http://ec.europa.eu/health/data_collection/data_protection/in_eu/index_en.htm. [Online; accessed Sempteber-2015].
- [41] “The court of justice declares that the commission’s us safe harbour decision is invalid.” <http://curia.europa.eu/jcms/upload/docs/application/pdf/2015-10/cp150117en.pdf>. [Online; accessed Sempteber-2015].
- [42] “Machine learning course.” <https://www.coursera.org/learn/machine-learning>. [Online; accessed September-2015].
- [43] “Machine learning.” <http://www.britannica.com/technology/machine-learning>. [Online; accessed September-2015].
- [44] “Why is machine learning (cs 229) the most popular course at stanford?.” <http://www.forbes.com/sites/anthonykosner/2013/12/29/why-is-machine-learning-cs-229-the-most-popular-course-at-stanford/>. [Online; accessed September-2015].
- [45] “Machine learning introduction.” http://www.aihorizon.com/essays/generalai/machine_learning.htm. [Online; accessed September-2015].
- [46] “Machine learning.” https://en.wikipedia.org/wiki/Machine_learning. [Online; accessed September-2015].
- [47] T. M. Mitchell, *Machine Learning*. New York, NY, USA: McGraw-Hill, Inc., 1 ed., 1997.

- [48] “Supervised and unsupervised learning.” <http://dataaspirant.com/2014/09/19/supervised-and-unsupervised-learning/>. [Online; accessed September-2015].
- [49] “Statistical classification.” https://en.wikipedia.org/wiki/Statistical_classification. [Online; accessed September-2015].
- [50] “Radial basis function network.” https://en.wikipedia.org/wiki/Radial_basis_function_network. [Online; accessed September-2015].
- [51] “Artificial neural network.” https://en.wikipedia.org/wiki/Artificial_neural_network. [Online; accessed September-2015].
- [52] “Weka - rbfnetwork.” <http://weka.sourceforge.net/doc/packages/RBFNetwork/weka/classifiers/functions/RBFNetwork.html>. [Online; accessed September-2015].
- [53] “Bayesian network.” https://en.wikipedia.org/wiki/Bayesian_network. [Online; accessed September-2015].
- [54] “Weka - bayesian network.” <http://weka.sourceforge.net/doc/dev/weka/classifiers/bayes/BayesNet.html>. [Online; accessed September-2015].
- [55] “Rotation forest: A new classifier ensemble method..” <http://www.ncbi.nlm.nih.gov/pubmed/16986543>. [Online; accessed September-2015].
- [56] “Ensemble learning.” https://en.wikipedia.org/wiki/Ensemble_learning. [Online; accessed September-2015].
- [57] “Class rotationforest.” <http://weka.sourceforge.net/doc/packages/rotationForest/weka/classifiers/meta/RotationForest.html>. [Online; accessed September-2015].
- [58] “Cohen’s kappa.” https://en.wikipedia.org/wiki/Cohen%27s_kappa. [Online; accessed September-2015].
- [59] “Receiver operating characteristic.” https://en.wikipedia.org/wiki/Receiver_operating_characteristic. [Online; accessed September-2015].
- [60] “Weka results explanation.” <http://stackoverflow.com/a/21551275/1933561>. [Online; accessed September-2015].

- [61] “Confusion matrix.” https://en.wikipedia.org/wiki/Confusion_matrix. [Online; accessed September-2015].
- [62] “Regression analysis.” https://en.wikipedia.org/wiki/Regression_analysis. [Online; accessed September-2015].
- [63] “Gaussian process.” https://en.wikipedia.org/wiki/Gaussian_process. [Online; accessed September-2015].
- [64] “Class smoreg.” <http://weka.sourceforge.net/doc.dev/weka/classifiers/functions/SMOreg.html>. [Online; accessed September-2015].
- [65] “Support vector machine.” https://en.wikipedia.org/wiki/Support_vector_machine. [Online; accessed September-2015].
- [66] “Cross-validation.” [https://en.wikipedia.org/wiki/Cross-validation_\(statistics\)](https://en.wikipedia.org/wiki/Cross-validation_(statistics)). [Online; accessed September-2015].
- [67] “Weka.” <http://www.cs.waikato.ac.nz/ml/weka/>. [Online; accessed September-2015].
- [68] “Jquery.” <https://jquery.com/>. [Online; accessed September-2015].
- [69] “Nodejs.” <https://nodejs.org/en/>. [Online; accessed September-2015].
- [70] “Mongodb.” <https://www.mongodb.org/>. [Online; accessed September-2015].
- [71] “Github.” <https://github.com/>. [Online; accessed September-2015].
- [72] “Heroku.” <http://heroku.com/>. [Online; accessed September-2015].
- [73] “Usage share of operating systems - desktop and laptop computers.” https://en.wikipedia.org/wiki/Usage_share_of_operating_systems#Desktop_and_laptop_computers. [Online; accessed September-2015].
- [74] “Usage share of web browsers.” https://en.wikipedia.org/wiki/Usage_share_of_web_browsers#Summary_tables. [Online; accessed September-2015].